

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Luka Ceraj

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Dubravko Majetić

Student:

Luka Ceraj

Zagreb, 2015.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu. Zahvaljujem se mentoru prof. dr. sc. Dubravku Majetiću na pomoći i savjetima prilikom same izrade završnog rada. Također, zahvaljujem se svojoj obitelji, rodbini i prijateljima na potpori koju su mi pružili tijekom studija.

Luka Ceraj



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
, materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	26-02-2015 Prilog
Klasa:	602-04/15-6/3
Ur.broj:	15-1703-15-81

ZAVRŠNI ZADATAK

Student: **Luka Ceraj**

Mat. br.: 0035173603

Naslov rada na
hrvatskom jeziku:

**Unaprijedna neuronska mreža s jednim i dva skrivena sloja u predikciji
ponašanja kaotičnog dinamičkog sustava**

Naslov rada na
engleskom jeziku:

**Feedforward Neural Network with One and Two Hidden Layers in
Prediction of Chaotic Dynamic System**

Opis zadatka:

Odabrati jedan ili dva skrivena sloja pri učenju statičke unaprijedne neuronske mreže problem je s kojim se susreću inženjeri kada krenu u planiranje topologije neuronske mreže za dani zadatak. U ovom se radu nastoji pokazati kvaliteta učenja i generalizacije mreža s unaprijed zadanim brojem neurona organiziranih u jednom ili dva skrivena sloja. Pri tome se koristi primjer učenja kaotičnog dinamičkog sustava.

U radu treba načiniti slijedeće:

1. Izvesti matematički model povratnog prostiranja pogreške statičke neuronske mreže s jednim i dva skrivena sloja neurona.
2. Radi dodatnog ubrzanja procesa učenja ugraditi metodu zamaha (momentuma) prvog i drugog reda.
3. Za primjer učenja predikcije ponašanja kaotičnog sustava definirati datoteku s podacima za učenje i nekoliko datoteka s podacima za testiranje mreže.
4. Obaviti postupak učenja i testiranja naučenih neuronskih mreža s jednim i dva skrivena sloja i jednakim ukupnim brojem skrivenih neurona. Pri tome neurone skrivenog sloja strukture s dva skrivena sloja organizirati u različite topologije neuronske mreže.
5. Programsku podršku načiniti u nekom od dostupnih matematičkih programskih paketa.
6. Izvesti zaključke rada.

Zadatak zadan:

25. studenog 2014.

Zadatak zadao:

Prof. dr. sc.  Dubravko Majetić

Rok predaje rada:

1. rok: 26. veljače 2015.

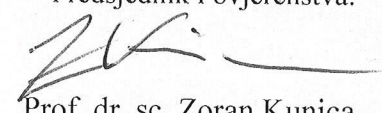
2. rok: 17. rujna 2015.

Predviđeni datumi obrane:

1. rok: 2., 3., i 4. ožujka 2015.

2. rok: 21., 22., i 23. rujna 2015.

Predsjednik Povjerenstva:


Prof. dr. sc. Zoran Kunica

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA	IV
POPIS OZNAKA	V
SAŽETAK	VII
SUMMARY	VIII
1. UVOD	1
1.1. Umjetne neuronske mreže	1
1.2. Biološki neuron	2
1.3. Sličnosti i razlike mozga i računala	4
1.4. Umjetni neuron	4
1.5. Vrste umjetnih neuronskih mreža	6
1.6. Učenje umjetnih neuronskih mreža	7
2. UNAPRIJEDNA STATIČKA NEURONSKA MREŽA S POVRATNIM PROSTIRANJEM POGREŠKE	9
2.1. Perceptron	9
2.2. Višeslojne neuronske mreže	11
2.2.1. Model statičkog neurona	11
2.2.2. Model statičke neuronske mreže	13
2.3. Učenje povratnim prostiranjem pogreške	14
2.3.1. Unaprijedna faza učenja statičke neuronske mreže	15
2.3.2. Povratna faza učenja statičke neuronske mreže	16
3. MATEMATIČKI IZVOD ZADANIH MODELA NEURONSKIH MREŽA	18
3.1. Neuronska mreža s jednim sakrivenim slojem	18
3.1.1. Unaprijedna faza učenja neuronske mreže s jednim sakrivenim slojem	18
3.1.1.1. Sakriveni sloj	18
3.1.1.2. Izlazni sloj	19
3.1.2. Povratna faza učenja neuronske mreže s jednim sakrivenim slojem	19
3.1.2.1. Promjena težina izlaznog sloja	20
3.1.2.2. Promjena težina sakrivenog sloja	21
3.2. Neuronska mreža s dva sakrivena sloja	23
3.2.1. Unaprijedna faza učenja neuronske mreže s dva sakrivena sloja	25
3.2.1.1. Prvi sakriveni sloj	25
3.2.1.2. Drugi sakriveni sloj	25
3.2.1.3. Izlazni sloj	26
3.2.2. Povratna faza učenja neuronske mreže s dva sakrivena sloja	26
3.2.2.1. Promjena težina izlaznog sloja	27
3.2.2.2. Promjena težina drugog sakrivenog sloja	28
3.2.2.3. Promjena težina prvog sakrivenog sloja	29
3.2.3. Primjer neuronske mreže s dva sakrivena sloja	32
3.3. Ocjena točnosti algoritma učenja	35

4. UNAPRIJEDNA STATIČKA NEURONSKA MREŽA ZA PREDVIĐANJE PONAŠANJA KAOTIČNOG DINAMIČKOG SUSTAVA	36
4.1. Glass-Mackeyev nelinearni kaotični sustav	36
4.2. Predviđanje ponašanja kaotičnog sustava	37
4.3. Učenje neuronske mreže	38
4.3.1. Učenje i testiranje neuronske mreže s jednim sakrivenim slojem	39
4.3.2. Učenje i testiranje neuronske mreže s dva sakrivena sloja	45
4.3.3. Usporedba učenja i testiranja neuronskih mreža s jednim i s dva sakrivena sloja	51
5. ZAKLJUČAK.....	53
LITERATURA.....	55

POPIS SLIKA

Slika 1.1.	Skica biološkog neurona.....	2
Slika 1.2.	Struktura umjetnog neurona.....	5
Slika 2.1.	Model statičkog neurona	11
Slika 2.2.	Prikaz nekih aktivacijskih funkcija neurona.....	12
Slika 2.3.	Struktura statičke unaprijedne neuronske mreže s jednim sakrivenim slojem	14
Slika 3.1.	Struktura statičke unaprijedne neuronske mreže s dva sakrivena sloja.....	24
Slika 3.2.	Struktura neuronske mreže za zadani primjer.....	32
Slika 4.1.	Prikaz Glass-Mackeyeve vremenske serije	37
Slika 4.2.	Prikaz podataka za učenje mreže	38
Slika 4.3.	Naučena mreža s jednim sakrivenim slojem.....	39
Slika 4.4.	Ostvareni NRMS mreže s jednim sakrivenim slojem	40
Slika 4.5.	Testiranje naučene mreže s jednim sakrivenim slojem na 500 novih i mreži u učenju nepoznatih točaka izlaza sustava	40
Slika 4.6.	Učenje mreže s jednim sakrivenim slojem uz zadanu sigmoidalnu aktivacijsku funkciju bez momentuma	41
Slika 4.7.	Učenje mreže s jednim sakrivenim slojem uz zadanu sigmoidalnu aktivacijsku funkciju i momentum prvog reda.....	41
Slika 4.8.	Učenje mreže s jednim sakrivenim slojem uz zadanu sigmoidalnu aktivacijsku funkciju i momentum drugog reda.....	42
Slika 4.9.	Učenje mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju bez momentuma	42
Slika 4.10.	NRMS mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju i momentum prvog i drugog reda.....	43
Slika 4.11.	Učenje mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju i momentum prvog reda uz koeficijent brzine učenja $\eta = 0,01$	43
Slika 4.12.	Učenje mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju i momentum drugog reda uz koeficijent brzine učenja $\eta = 0,01$	44
Slika 4.13.	Naučena mreža s dva sakrivena sloja	45
Slika 4.14.	Ostvareni NRMS mreže s dva sakrivena sloja.....	46
Slika 4.15.	Testiranje naučene mreže s dva sakrivena sloja na 500 novih i mreži u učenju nepoznatih točaka izlaza sustava.....	46
Slika 4.16.	Učenje mreže s 2 neurona prvog sloja i 18 neurona drugog sloja	48
Slika 4.17.	Učenje mreže s 5 neurona prvog sloja i 15 neurona drugog sloja	48
Slika 4.18.	Učenje mreže s 8 neurona prvog sloja i 12 neurona drugog sloja	49
Slika 4.19.	Učenje mreže s jednakim brojem neurona sakrivenih slojeva.....	49
Slika 4.20.	Učenje mreže s 12 neurona prvog sloja i 8 neurona drugog sloja	49
Slika 4.21.	Učenje mreže s 15 neurona prvog sloja i 5 neurona drugog sloja	50
Slika 4.22.	Učenje mreže s 18 neurona prvog sloja i 2 neurona drugog sloja	50
Slika 4.23.	Usporedba učenja mreža s jednim i s dva sakrivena sloja.....	52
Slika 4.24.	Usporedba testiranja mreža s jednim i s dva sakrivena sloja.....	52

POPIS TABLICA

Tablica 4.1.	Usporedba učenja mreža s jednim sakrivenim slojem s različitim kombinacijama aktivacijskih funkcija i momentuma.....	44
Tablica 4.2.	Usporedba učenja mreža s dva sakrivena sloja za različite kombinacije aktivacijskih funkcija i momentuma	47
Tablica 4.3.	Usporedba učenja mreža s dva sakrivena sloja za različite kombinacije broja neurona sakrivenih slojeva	51
Tablica 4.4.	Usporedba učenja mreža s jednim i s dva sakrivena sloja.....	51

POPIS OZNAKA

BIAS – neuron bez ulaza konstantnog izlaza jednakog jedan

d – željeni izlazi mreže

E – funkcija cilja (suma kvadrata pogrešaka)

I – broj ulaznih neurona + BIAS

J – broj neurona sakrivenog sloja + BIAS

K – broj izlaznih neurona

L – broj neurona drugog sakrivenog sloja + BIAS

NRMS – normalizirani korijen srednje kvadratne pogreške (mjera točnosti)

net – suma umnožaka ulaza neurona i pripadajućih težinskih faktora

O – izlazi neuronske mreže

u – težinski faktori koji povezuju izlaze neurona prvog sakrivenog sloja s ulazima neurona drugog sakrivenog sloja

v – težinski faktori koji povezuju izlaze ulaznih neurona s ulazima neurona sakrivenog sloja

w – težinski faktori koji povezuju izlaze neurona sakrivenog sloja s ulazima neurona izlaznog sloja

y – izlazna vrijednost neurona

Z – ulazi neuronske mreže

α – vrijednost koeficijenta momentuma prvog reda

β – vrijednost koeficijenta momentuma drugog reda

γ – aktivacijska funkcija neurona

η – koeficijent brzine učenja

δ – odstupanje izračunate izlazne vrijednosti od željene izlazne vrijednosti

SAŽETAK

Klasične statističke metode obrade signala daju slabe rezultate u predviđanju ponašanja dinamičkih sustava, a poseban problem predstavljaju sustavi s nelinearnostima, vremenski promjenjivim parametrima i kaotični sustavi. Promjenom parametara kaotični sustavi pokazuju širok raspon različitih nelinearnih ponašanja, te zato predstavljaju dobar problem za ispitivanje raznih tehnika obrade signala, među kojima su i neuronske mreže. U ovom radu biti će prikazan jedan takav problem, gdje će se pomoću unaprijedne statičke neuronske mreže s povratnim prostiranjem pogreške pokušati predvidjeti ponašanje nelinearnog kaotičnog sustava pod nazivom Glass-Mackeyeva jednadžba. Usporediti će se rezultati učenja neuronske mreže s jednim i s dva sakrivena sloja, ispitati utjecaj različitih aktivacijskih funkcija sakrivenih slojeva na učenje mreže, te utjecaj momentuma prvog i drugog reda. Prikazati će se i utjecaj broja neurona pojedinog sakrivenog sloja neuronske mreže s dva sakrivena sloja na učenje mreže. Naučena mreža biti će testirana na njoj nepoznatom skupu ulaznih podataka.

Ključne riječi:

predikcija

kaotični sustavi

unaprijedna statička neuronska mreža

povratno prostiranje pogreške

Glass-Mackeyev sustav

sakriveni slojevi

aktivacijske funkcije

momentum prvog reda

momentum drugog reda

SUMMARY

Classical statistical signal processing methods give poor results in predicting the behavior of dynamic systems and special issue represent systems with nonlinearity, time-varying parameters and chaotic systems. By changing their parameters chaotic systems show a wide range of different nonlinear behavior and therefore represent a good problem for testing a variety of signal processing techniques, including the neural networks. This paper will be presented with one such problem, where Error-Back Propagation Neural Network with Static Neuron Models will be used in prediction of the behavior of nonlinear chaotic system called Glass-Mackey equation. Learning results of neural networks with one and two hidden layers will be compared, impact of different activation functions of hidden layers on network learning will be examined, and the impact of the momentum of the first and second order. Influence of the number of neurons in each hidden layer of neural network with two hidden layers on learning of the network will be shown. Trained network will be tested with input data not known to network.

Key words:

prediction

chaotic systems

feedforward static neural network

error-back propagation

Glass-Mackey system

hidden layers

activation functions

first order momentum

second order momentum

1. UVOD

Umjetne neuronske mreže su elementi umjetne inteligencije. Originalna ideja za umjetnu neuronsku mrežu nastala je iz niza pokušaja modeliranja biofiziologije mozga čovjeka, s ciljem razumijevanja i objašnjenja kako mozak funkcionira. Cilj ideje da se pokuša modelirati biofiziologija ljudskog mozga je stvaranje modela sposobnog da procesira, tj. prihvata, obrađuje, generira, pohranjuje i prenosi informacije, analogno aktivnostima mozga čovjeka.

1.1. Umjetne neuronske mreže

Godine 1890. u svojoj knjizi William James [2] iznosi tvrdnju koja je poslužila za izgradnju osnovne strukture umjetnog neurona: Aktivnost bilo koje točke mozga čovjeka predstavlja zbroj tendencija svih ostalih točaka da se „prazne“ tj. ispaljuju u nju. Te tendencije proporcionalne su:

- a) broju točaka (broju veza) koje djeluju na promatranu točku
- b) intenzitetu tih uzbuda (težini veza)
- c) odsutnosti rivalne točke koja nije u funkcionalnoj vezi s promatranom točkom, a u koju bi se „pražnjenje“ ostalih točaka moglo skrenuti.

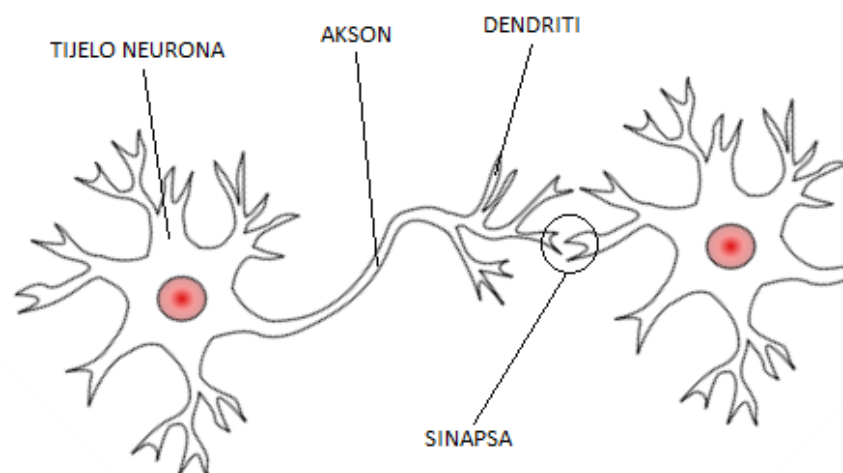
Na temelju te tvrdnje, ako se određena točka u mozgu čovjeka zamijeni neuronom, onda se aktivnost neurona može modelirati kao zbroj „otežanih“ ulaza neurona. Ti otežani ulazi su ulazi pomnoženi određenim faktorima koji se nazivaju težine neurona. Prema tome aktivnost umjetnog neurona ovisi o broju ulaza (veza) iz okoline (okruženja) neurona, intenzitetu tih veza (iznosu težinskih faktora), te o pragu osjetljivosti koji stanje neurona mora dosegnuti prije nego što „ispali impuls“ preko svog izlaza u okolinu neurona. Okruženje neurona čine ostali neuroni umjetne neuronske mreže i/ili okruženje te mreže [1].

1943. godine Warren McCulloch i Walter Pitts napravili su formalizaciju aktivnosti umjetnog neurona osnovanoj na tvrdnji Williama Jamesa, predloživši jednostavan model umjetnog neurona [3]. Takav model umjetnog neurona i danas služi kao osnova za izgradnju umjetnih neuronskih mreža. Iako je sam model isti za sve umjetne neuronske mreže, one se međusobno razlikuju po strukturi veza između neurona i samog neurona s okruženjem mreže te po načinu određivanja intenziteta tih veza (određivanja težina mreže), što predstavlja

proces učenja mreže. Stvaranje umjetnih neuronskih mreža može se podijeliti na dva, po principu različita pristupa, ali je teško odrediti točnu granicu među njima, jer se u oba pristupa koriste dosta zajedničkih elemenata. Prvi pristup ima za cilj realizirati modele koji dovoljno točno oponašaju aktivnosti mozga čovjeka, što bi trebalo rezultirati sustavom umjetne inteligencije. Drugi pristup podrazumijeva razvoj umjetnih neuronskih mreža s velikim računarskim sposobnostima, poput neuroračunala koja su namijenjena za rješavanje konkretnih zadataka praktične prirode. Peter J. W. Melsi termin „umjetne neuronske mreže“ koristi kako bi se opisale različite topologije u visokom stupnju međusobno povezanih jednostavnih procesnih elemenata (neurona), koje nude alternativu (paralelno procesiranje) tradicionalnom pristupu računarstvu (serijska von Neumannova računala). Osnovne prednosti neuroračunala nad klasičnim (serijskim) računalima su paralelno procesiranje informacija, adaptabilnost, te sposobnost učenja. Umjetna neuronska mreža se stoga, može definirati kao složeni sustav, sastavljen od elemenata (neurona), koji stoje u određenoj interakciji, međusobno i s okolinom sustava (težine neuronske mreže) tako da grade funkcionalnu cjelinu [1]. Zbog usporedbe s umjetnim neuronom u nastavku će biti izložen kratak opis biološkog neurona.

1.2. Biološki neuron

Biološki neuron [1, 4] se sastoji od tijela neurona, aksona i dendrita koji okružuju tijelo neurona. Na slici 1.1. prikazana je pojednostavljena građa neurona:



Slika 1.1. Skica biološkog neurona

Akson je tanka cjevčica koja je na jednom kraju povezana s tijelom neurona, a drugi kraj se dijeli na niz grana. Krajevi tih grana završavaju malim zadebljanjima koji najčešće dodiruju dendrite, a rjeđe tijelo drugog neurona. Mali razmak između kraja aksona prethodnog neurona i dendrita ili tijela sljedećeg neurona naziva se sinapsa. Akson jednog neurona stvara sinaptičke veze s mnoštvom drugih neurona. Impulsi (izlazi) neurona putuju kroz akson do sinapsi odakle se signali različitog intenziteta šalju kroz dendrite ili direktno na tijelo drugog neurona. Intenzitet impulsa je određen efikasnošću svakog pojedinačkog sinaptičkog prijenosa. Ako je neuron doveden u stanje dovoljne uzbuđene poslat će impuls kroz svoj akson, realizirano njegovim trenutačnim (početnim) stanjem te utjecajem signala od ostalih neurona preko njegovih dendrida ili njegovog tijela u kratkom vremenskom intervalu (period latentne sumacije). Signali koje prenose dendridi mogu biti smirujući (sprječavajući, kočeći, inhibirajući) ili uzbuđni (matematički gledano suprotnog su predznaka). Neuron će „ispaliti“, odnosno poslati impuls kroz svoj akson ako je njegova uzbuđena veća od smirujućeg utjecaja za kritični iznos, koji predstavlja prag osjetljivosti neurona.

Biološka neuronska mreža je vrlo složena. Procjenjuje se da ljudski mozak ima preko 100 milijardi neurona, koji komuniciraju kroz tijelo preko neuronskih vlakana, te je moguće da grade i do 100 trilijuna veza. Ta mreža je odgovorna za sve procese koje zovemo razmišljanje, učenje, emocije, spoznaja, percepcija, ali i za izvođenje odgovarajuće motorne senzoričke autonomnih funkcija poput rada srca, disanja, refleksa itd. Isto tako dodatnoj složenosti cijele biološke neuronske mreže pridonosi i činjenica da samo jedan neuron može generirati impuls koji će aktivirati ili smiriti na tisuće drugih neurona, iako ti isti neuroni mogu biti uzbuđivani od tisuća drugih neurona. Ovim se realizira visok stupanj povezanosti neurona u neuronskoj mreži, što upućuje na zaključak da se kompleksne funkcije neuronske mreže ostvaruju kompleksnošću veza među neuronima, a ne isključivo kompleksnošću svakog neurona zasebno.

Iako je veliki dio funkcionalnih aktivnosti biološke neuronske mreže već objašnjen, egzaktni način rada mreže nije dovoljno razumljiv. Ovisno o obliku tijela, koje je reda veličine 5 do 100 mikrona u promjeru, identificirano je nekoliko stotina bioloških neurona. Različiti oblici tijela neurona pripisuju se specijalizaciji neurona za izvođenje određenih važnih funkcija. Komunikacija u mozgu se odvija na dva načina: kemijskim signalima preko sinapsi i električnim signalima unutar neurona. Ovu izvanredno složenu aktivnost kreiranja obje vrste signala omogućava celularna membrana. Brzina ispaljivanja neurona je određena

kumulativnim učinkom velikog broja uzbuđujućih i smirujućih ulaza, koji se obrađuju u tijelu neurona u kratkom vremenskom intervalu. Uzbuđujući ulazi povećavaju, a smirujući smanjuju brzinu ispaljivanja neurona. Na taj način neuronski signali postaju frekvencijski modulirani (FM) impulsi.

1.3. Sličnosti i razlike mozga i računala

Ljudski mozak i računalo imaju nekoliko sličnosti. Oba koriste električne signale, imaju veliki broj jednostavnih elemenata te izvode funkcije koje se općenito mogu nazvati računarskim funkcijama. Isto tako, postoje i razlike. Za rješenje nekog problema računalo se programira dok mozak uči. Brzina prijenosa neuronskih impulsa je reda veličine milisekunde, a kod novijih, klasičnih računala je reda veličine nanosekunde. Iako računala puno brže prenose podatke, mozak je neusporedivo brži što se tiče računanja, zahvaljujući ogromnom broju računarskih jedinica (neurona), koje za razliku od računala koja rade serijski (sljedeća sekvenca se izvršava tek kada prethodna završi), rade paralelno, što značajno ubrzava ukupni proces računanja [1].

Bitna razlika između mozga i računala je stupanj, tj. kvaliteta pogreške. Računalo ne radi pogrešku tako dugo dok su mu svi ulazi, softver i hardver savršeno ispravni. Mozak daje bolje zaključke i aproksimacije čak i za djelomično nepotpune i neispravne ulaze.

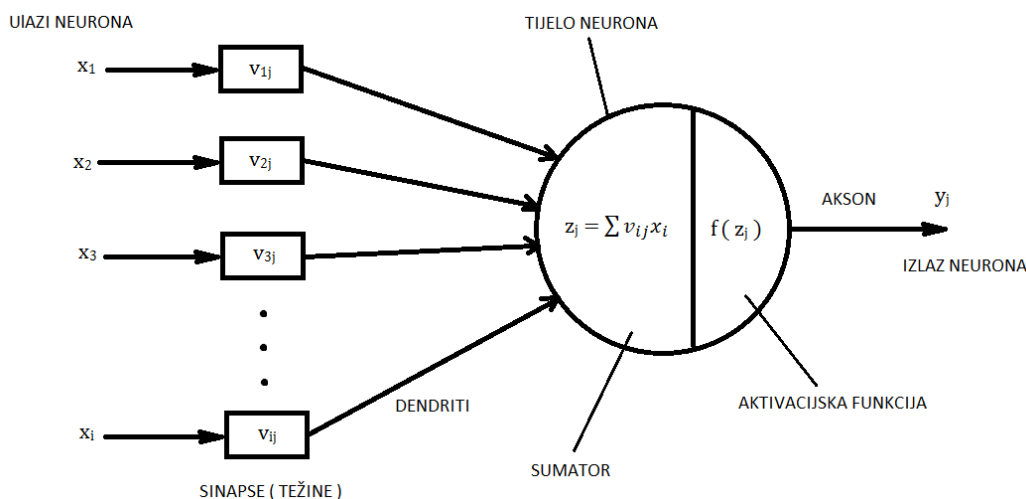
Potrošnja energije također je velika razlika između mozga i računala [1]. Mozak je najkoncentriraniji potrošač energije u tijelu čovjeka. Masa mu je svega 2 % mase cijelog tijela, ali zato troši preko 20 % ukupnog kisika koji je potreban tijelu. Ako se potrošnja mozga izrazi kao energija potrošena u određenom vremenu, ispada da mozak troši samo 20 W energije, što ga čini izuzetno uspješnim energetske članom. Za usporedbu, računalo koje bi imalo identičan kapacitet kao mozak čovjeka, trošilo bi na tisuće vati energije.

Očito je da su dva navedena sustava različita u mnogim stvarima. Optimirana su da rješavaju različite tipove problema, imaju temeljno različitu strukturu i razvijana su uz bitno različite kriterije.

1.4. Umjetni neuron

Umjetni neuron [1, 5] se dizajnira s idejom da oponaša osnovne funkcije biološkog neurona. Sumator umjetnog neurona zamjenjuje tijelo biološkog neurona dok ulogu dendrita

preuzimaju ulazi u sumator, a izlaz iz sumatora je akson umjetnog neurona. Prag osjetljivosti umjetnog neurona je izveden pomoću aktivacijske funkcije te ona simulira prag osjetljivosti biološkog neurona. Ostvarivanje veze umjetnog neurona sa njegovom okolinom postiže se pomoću težinskih faktora koji preslikavaju funkcije sinaptičke veze biološkog neurona. Na slici 1.2. je prikazana struktura umjetnog neurona.



Slika 1.2. Struktura umjetnog neurona

Težinski faktor može biti pozitivan ili negativan broj, a može biti i neka funkcija pa je težinski faktor varijabilan. Ukoliko je težinski faktor jednak nuli, odgovarajuća veza s okolinom neurona ne postoji, pa se u shemi neuronske mreže ni ne ucrtava. Težinski faktori imaju istu ulogu kao i sinapse kod bioloških neurona. Oni povezuju izlaze iz okoline neurona, tj. izlaze drugih neurona (aksone) s ulazima sumatora (dendriti). Intenzitet te veze ovisi o iznosu, a karakter veze o predznaku težinskog faktora. Izlaz sumatora povezuje se na ulaz aktivacijske funkcije, koja na svom izlazu daje izlaz umjetnog neurona. Aktivacijske funkcije mogu biti linearne i nelinearne. Kod linearnih aktivacijskih funkcija se izlaz sumatora množi sa nekim faktorom, tj. pojačanjem te se tako dobiva izlaz neurona. Nelinearne aktivacijske funkcije mogu poprimiti različite oblike, ali najčešće se koriste funkcije praga osjetljivosti, sigmoidalne, hiperbolične i harmoničke funkcije. Nelinearne aktivacijske funkcije prevode izlaz sumatora na izlaz neurona preko nelinearnog pojačanja. Tako funkcija praga osjetljivosti daje na izlazu neurona jedinicu ako je izlaz sumatora veći od zadanog praga osjetljivosti, što odgovara ispaljivanju impulsa kod biološkog neurona. U suprotnom slučaju izlaz neurona je

nula te je on neaktivan.

Može se zaključiti da umjetni neuron funkcionira slično kao i biološki neuron. Izlazi iz drugih neurona i/ili okruženja promatranog neurona, koji se upućuju promatranom neuronu, množe se s težinskim faktorima i dovode do sumatora. U sumatoru se tako dobiveni produkti sumiraju, a njihova suma se odvodi na ulaz aktivacijske funkcije koja će na svom izlazu dati izlaz neurona. Opisani umjetni neuron se često naziva prema svojim autorima McCulloch-Pittsiv neuron. Ukoliko je aktivacijska funkcija oblika skok funkcije takav neuron se u literaturi često naziva Perceptron. Jednostavna struktura opisanog umjetnog neurona zanemaruje mnoge karakteristike biološkog neurona. Tako, na primjer, ne vodi računa o kašnjenju signala (ulaz neurona producira trenutačno njegov izlaz), što se odražava na dinamiku sustava, iako to u nekim slučajevima može biti prednost, jer se povećava brzina procesiranja ulaznih signala mreže. Jednako tako opisani umjetni neuron ne uključuje efekte sinkronizacije, odnosno frekventne modulacije biološkog neurona. Sam umjetni neuron nije u mogućnosti realizirati funkcije, ali je to moguće postići odgovarajućim povezivanjem neurona u umjetnu neuronsku mrežu. Bez obzira na ograničenja opisanog umjetnog neurona, umjetne neuronske mreže realizirane povezivanjem takvih neurona mogu dati dobre rezultate.

1.5. Vrste umjetnih neuronskih mreža

Umjetne neuronske mreže je moguće podijeliti prema više kriterija [1, 6]. Prema broju slojeva mogu biti jednoslojne i višeslojne. Paralelno složen skup neurona gradi jedan sloj neuronske mreže. Ukoliko se radi o višeslojnim mrežama uobičajeno je da te mreže imaju ulazni i izlazni sloj, a između njih se nalazi skriveni sloj. Ako se slojevi mreže povežu tako da signal putuje samo u jednom smjeru, takve mreže se nazivaju unaprijedne (eng. feedforward neural networks), a ako postoji barem jedna povratna petlja, kroz koju se signal vraća na ulaz, mreže su povratne (eng. feedback neural networks). Prilikom rješavanja različitih problema bitno je koje od tih mreža će se koristiti, jer se neki problemi mogu riješiti unaprijednim neuronskim mrežama, dok se drugi problemi mogu riješiti pomoću povratnih neuronskih mreža. Uvođenjem povratne petlje u neuronsku mrežu uključuje se i vrijeme. Kod povratnih neuronskih mreža izlaz nekog neurona može djelovati povratno na samog sebe i na druge neurone. Nemoguće je da se to dogodi istovremeno, jer se najprije mora generirati izlaz neurona da bi mogao povratno djelovati na samog sebe ili ostale neurone. Samim time operacije signala u sumatoru kao i signali postaju funkcije vremena. U odnosu na vrijeme

razlikuju se vremenski kontinuirane i vremenski diskretne neuronske mreže.

Postoji niz drugih podjela neuronskih mreža [1]:

- ovisno o glavnom području primjene postoje perceptronske, asocijativne, dvostruko asocijativne, adaptivne, kognitrone i neokognitrone mreže,
- ovisno o metodama koje se koriste za učenje mreža imamo povratno propagirane, suprotno propagirane i statističke mreže,
- prema načinu realiziranja postoje softverske, hardverske i optičke neuronske mreže.

1.6. Učenje umjetnih neuronskih mreža

Učenje neuronskih mreža se dijeli na supervizorno i nesupervizorno učenje [1, 6]. Supervizorno učenje zahtjeva vanjskog učitelja neuronske mreže, koji promatra ponašanje mreže te na temelju toga istu i ispravlja kako bi se postiglo željeno ponašanje mreže. Prilikom takvog načina učenja neuronskih mreža, prvo je potrebno odrediti broj ulaza, broj slojeva, broj izlaza, te broj težina mreže kako bi se definirala struktura neuronske mreže. Početne težine obično se usvajaju generiranjem slučajnih vrijednosti. Zatim se na ulaz mreže dovodi skup ulaznih varijabli te mreža producira skup izlaznih varijabli. Izlazne varijable se potom uspoređuju sa željenim izlaznim varijablama. Razlika željenih i stvarnih izlaza gradi pogrešku mreže, koja se zatim koristi za računanje novih težina mreže preko usvojenog algoritma. Cijeli postupak se ponavlja iterativno dok pogreška mreže ne bude manja od unaprijed zadanog iznosa. Prilikom učenja je po potrebi moguće mijenjati i strukturu mreže (broj neurona, slojeva i težina). Nakon procesa učenja slijedi proces testiranja neuronske mreže. Testiranje se provodi sa novim skupom ulaza koji nije sadržan u ulaznim podacima tijekom učenja. Prilikom testiranja mreža producira nove izlaze koji se uspoređuju sa željenim izlazima, a pritom se ne mijenjaju parametri mreže. Iznos pogreške u procesu testiranja služi za ocjenu robusnosti mreže, tj. sposobnosti mreže da daje zadovoljavajuće izlaze i za skup ulaza koji nisu korišteni u fazi učenja.

Za razliku od supervizornog učenja, nesupervizorno učenje neuronske mreže ne koristi vanjskog učitelja, nego se mreža sama organizira. Mreže učene ovom metodom još se nazivaju i samoorganizirajuće neuronske mreže. Na ulaz mreže se dovodi skup ulaznih varijabli, a mreža po dobro definiranom algoritmu samoorganizira podešavanjem svojih parametara. Obzirom da željeni izlaz nije definiran za vrijeme učenja, rezultat učenja nije

moгуće predvidjeti. Tako organizirana mreža se mođe koristiti za klasifikaciju uzoraka, odnosno njihovo prepoznavanje. Kao i kod supervizornog učenja nakon procesa učenja se također provodi proces testiranja mreže.

Postoji mnoštvo različitih algoritama učenja neuronskih mreža, a neki od njih su: generalizirano delta pravilo, povratno propagiranje, optimalno učenje, statističko i adaptivno učenje te drugi.

2. UNAPRIJEDNA STATIČKA NEURONSKA MREŽA S POVRATNIM PROSTIRANJEM POGREŠKE

U nastavku će biti prikazane osnovne karakteristike unaprijednih neuronskih mreža. Kako bi se objasnilo funkcioniranje unaprijednih mreža, uz pomoć modela perceptrona, biti će objašnjeno delta pravilo učenja, a zatim će biti objašnjeno učenje povratnim prostiranjem pogreške.

2.1. Perceptron

Perceptron je binarni umjetni neuron čiji izlaz može poprimiti vrijednosti 0 ili 1. Kod perceptrona kao aktivacijska funkcija se koristi binarna funkcija praga osjetljivosti koja je definirana kao:

$$y_j = f(z_j). \quad (2.1)$$

Vrijednost izlaza će biti 1 ako suma ulaznih vrijednosti z_j bude veća od praga osjetljivosti, te će neuron biti aktivan, a ako vrijednost z_j bude manja ili jednaka pragu osjetljivosti, izlaz neurona će biti 0 i on će biti neaktivan [1].

Ako je perceptron aktivan, šalje signal susjednim neuronima s kojima je povezan u mrežu. Najjednostavniji model mreže je onaj s jednim ulaznim i jednim izlaznim slojem neurona, koji se naziva jednoslojna perceptronska mreža. Mreža može imati više slojeva, gdje su neuroni jednog sloja povezani s neuronima prethodnog i sljedećeg sloja, ali neuroni istog sloja nisu međusobno povezani. Ulazni sloj služi za prijenos signala, preko skrivenih, do izlaznog sloja. Ulazi svi neurona, osim onih ulaznog sloja, se množe s pridruženim težinama i sumiraju. Težine su podesivi parametri perceptronskih mreža, te su predmet učenja. Učenje se provodi pomoću skupa za učenje, koji se sastoji od većeg broja uzoraka opisanih ulaznim vektorima i njihovih željenih izlaznih vrijednosti. Cilj učenja je podešavanje nepoznatih vrijednosti težina na način da izlazni neuron koji pripada istoj skupini kao i uzorak skupa za učenje poprimi vrijednost 1, a svi ostali neuroni vrijednost 0.

Perceptronske mreže uče iterativnim postupkom na sljedeći način:

1. Za ulazni vektor x uzorka iz skupa za učenje izračuna se izlazni vektor y :

$$y = x W \quad (2.2)$$

gdje je W matrica težina, čiji se elementi u prvom koraku najčešće inicijaliziraju kao slučajni brojevi u intervalu od 0 do 1.

2. a) Ako je izlaz točan, povrat na točku 1.
 - b) Ako je izlaz pogrešan, i ako je 0 dodaju se vrijednosti ulaznog vektora pripadajućim težinama.
 - c) Ako je pogrešan, i ako je 1 od pripadajućih težina oduzimaju se vrijednosti ulaznog vektora.
3. Povrat na točku 1.

Postupak učenja iterativno se provodi za sve uzorke skupa za učenje tako dugo dok svi nisu pravilno razvrstani u pripadajuće skupine [1].

Generalizacijom postupka učenja perceptrona, za neurone s kontinuiranim ulaznim i izlaznim vrijednostima, te kontinuiranom aktivacijskom funkcijom, dobiveno je delta pravilo, koje je osnova za učenje većine modela neuronskih mreža:

$$\delta_j = T_j - A_j \quad (2.3)$$

gdje je T_j željena izlazna vrijednost, a A_j izračunata izlazna vrijednost izlaznog neurona j . Razlika između te dvije vrijednosti, δ_j naziva se odstupanje. Podesive težine w_{ij} mijenjaju se prema sljedećim izrazima:

$$\Delta_{ij} = \eta \delta_j x_i \quad (2.4)$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij} \quad (2.5)$$

gdje su:

- i – indeks neurona ulaznog sloja
- j – indeks neurona izlaznog sloja
- Δw_{ij} – faktor promjene težina
- η – koeficijent brzine učenja
- $w_{ij}(n+1)$ – težina veze između neurona i ulaznog sloja i neurona j izlaznog sloja poslije podešavanja
- $w_{ij}(n)$ – težina veze između neurona i ulaznog sloja i neurona j izlaznog sloja prije podešavanja

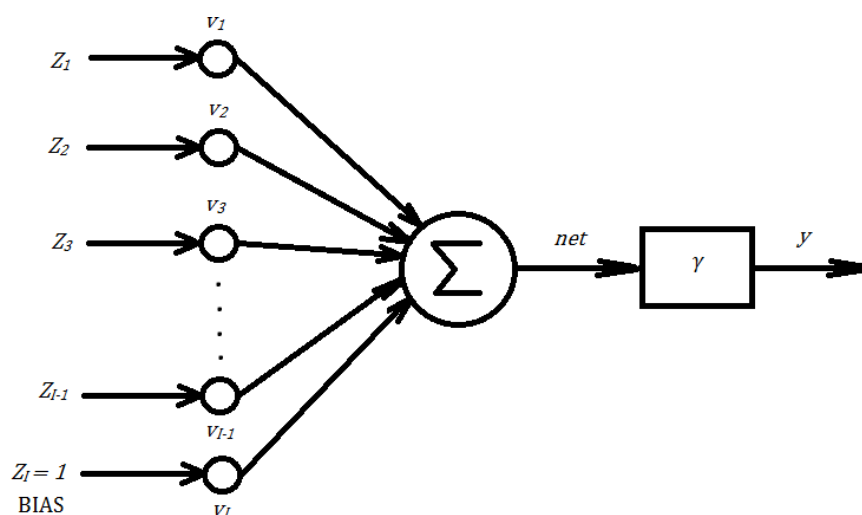
Veliki nedostatak jednoslojnih perceptronskih mreža je nemogućnost rješavanja velikog broja jednostavnih problema kod kojih su uzorci linearno neseparabilni [1]. Taj problem se može riješiti uvođenjem dodatnih slojeva u mrežu, uz uvjet da se u njima koristi nelinearna aktivacijska funkcija. Zadatak tih skrivenih slojeva je razdvajanje linearno neseparabilnih uzorka. Dodatan problem je učenje težina skrivenog sloja, budući da se delta pravilo u formi prikazanoj izrazima (2.4) i (2.5) ne može primijeniti na neurone skrivenog sloja, pošto za njih nije poznat željeni vektor.

2.2. Višeslojne neuronske mreže

Slijedi prikaz najčešćeg tipa neuronskih mreža, diskretne statičke neuronske mreže s povratnim rasprostranjem pogreške (eng. Error-Back Propagation Neural Network with Static Neuron Models). Osnovni element takve mreže je statički neuron.

2.2.1. Model statičkog neurona

Statički neuron [1, 5] posjeduje više ulaza i samo jedan izlaz, što je njegova osnovna značajka. Ima dvije temeljne funkcije, funkciju sume i aktivacijsku funkciju. Model statičkog neurona prikazan je na slici 2.1.



Slika 2.1. Model statičkog neurona

Kako bi se učenje mreže moglo uspješno odvijati, svaki neuron koji sudjeluje u procesu učenja ima poseban ulaz jedinične vrijednosti koji je u strukturi neuronske mreže

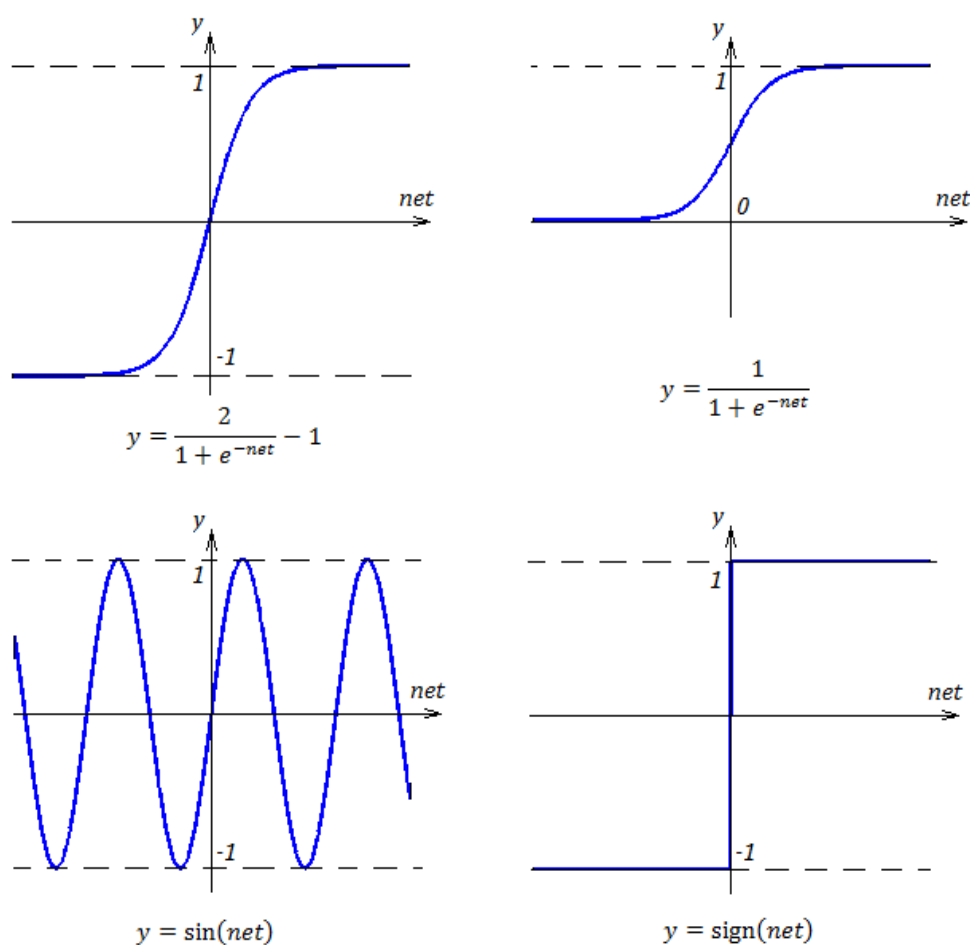
realiziran vezom s zasebnim neuronom oznake BIAS, koji ima konstantan izlaz vrijednosti 1. Funkcija sume statičkog neurona je suma umnožaka ulaza neurona i pripadajućih težinskih faktora i rezultira vrijednošću net :

$$net = \sum_{i=1}^l v_i Z_i . \quad (2.6)$$

Aktivacijska funkcija γ vrši preslikavanje vrijednosti net u izlaznu vrijednost neurona y :

$$y = \gamma (net) . \quad (2.7)$$

Za aktivacijsku funkciju najčešće se koriste monotono rastuće funkcije sa zasićenjem. Odabir aktivacijske funkcije neurona ovisi o skupu uzoraka za učenje i skupu uzoraka za testiranje naučene neuronske mreže. Primjeri nekih aktivacijskih funkcija prikazani su na slici 2.2.



Slika 2.2. Prikaz nekih aktivacijskih funkcija neurona

Najčešće korištena aktivacijska funkcija neurona je nelinearna bipolarna sigmoidalna funkcija prikazana na slici 2.2. u prvom dijagramu. Ta aktivacijska funkcija je globalnog karaktera i pokazuje dobre rezultate i kad se koristi u učenju dinamičkih neuronskih mreža. U ovom radu će biti prikazano učenje pomoću nelinearne bipolarne sigmoidalne funkcije i sinusne funkcije. Ako se odabere sigmoidalna funkcija, izraz (2.7) poprima sljedeći oblik:

$$y = \frac{2}{1+e^{-net}} - 1 \quad (2.8)$$

Ako se odabere sinusna funkcija, izraz (2.7) poprima sljedeći oblik:

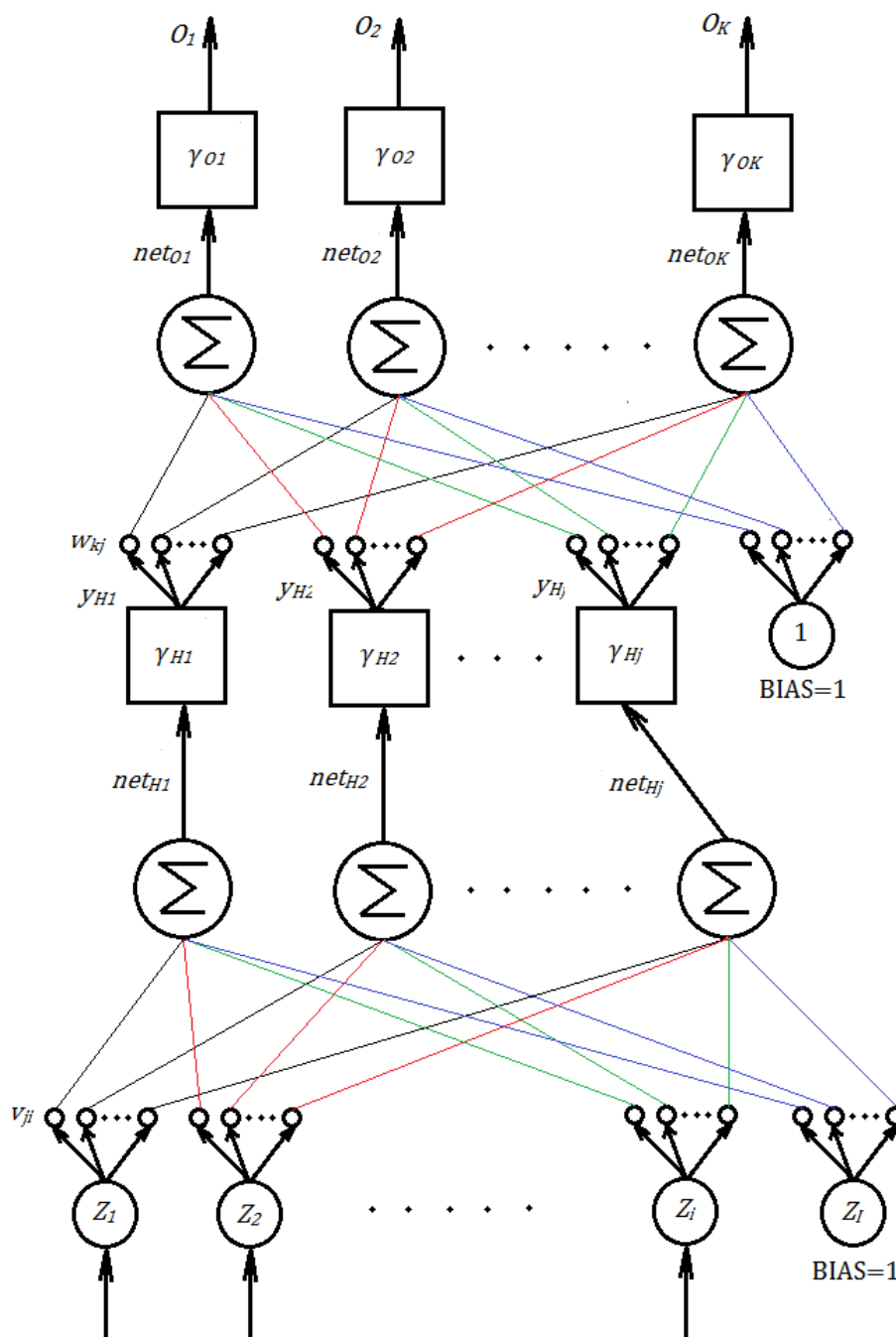
$$y = \sin (net) \quad (2.9)$$

2.2.2. Model statičke neuronske mreže

Model statičke neuronske mreže [1, 6] je sastavljen od niza neurona koji su raspodijeljeni u slojeve. Slojevi mreže su međusobno povezani težinskim faktorima. Slojevi mreže se dijele na ulazni, sakriveni i izlazni sloj. Ulazni i izlazni sloj su u direktnoj vezi s okolinom, a sakriveni slojevi nisu. Na slici 2.3. je prikazana struktura statičke unaprijedne neuronske mreže s jednim sakrivenim slojem.

Prema slici 2.3. ulazi neurona ulaznog sloja Z_i su ustvari ulazi u mrežu. Neuroni u ulaznom sloju zapravo nisu pravi neuroni, nego su čvorovi koji služe za distribuciju ulaza neuronske mreže prema prvom sakrivenom sloju neurona. Ulazni sloj je sa sakrivenim slojem povezan vezama opterećenim težinama v_{ji} . Svi slojevi mreže su potpuno povezani, tj. svaki neuron promatranog sloja je povezan sa svakim neuronom prethodnog sloja. Jedini izuzetak su neuroni s oznakom BIAS, koji imaju konstantnu izlaznu vrijednost jednaku 1.

Broj sakrivenih slojeva je proizvoljan, a najčešće se koristi jedan ili dva sakrivena sloja neurona. Broj neurona u sakrivenom sloju je proizvoljan, tj. ovisi o zadatku te se određuje eksperimentalno. Broj ulaznih i izlaznih neurona je uvijek definiran karakterom preslikavanja koje mreža mora obaviti. Primjerice ako se radi o učenju ponašanja nekog dinamičkog sustava, broj ulaznih neurona odgovara broju ulaza sustava, a broj izlaznih neurona odgovara broju izlaza sustava. Izlazni sloj neurona definira izlaze neuronske mreže O_k , a povezan je sa sakrivenim slojem neurona težinskim koeficijentima označenim sa w_{kj} .



Slika 2.3. Struktura statičke unaprijedne neuronske mreže s jednim sakrivenim slojem

2.3. Učenje povratnim rasprostranjem pogreške

Postupak učenja neuronske mreže nije ništa drugo nego podešavanje težinskih parametara učenja među slojevima mreže sa ciljem da se izlazi mreže, za odgovarajuće ulaze skupa za učenje, što više približe poznatim, željenim vrijednostima izlaza mreže. Iz toga se može zaključiti da postupak učenja ne može osigurati potpunu točnost. Izlaz mreže uvijek je

aproksimacija željenog izlaza. Kvaliteta te aproksimacije je u funkciji više čimbenika kao što su zadatak učenja, odabrana struktura mreže, te odabrani algoritam učenja. Algoritmi učenja se mogu podijeliti u dvije osnovne skupine: algoritme jednog koraka učenja i iteracijske algoritme učenja. Osnovna karakteristika učenja s učiteljem je da se u iterativnom postupku učenja, tj. iterativnom postupku promjene težinskih koeficijenata veza, mreži uzastopce prikazuju ulazne veličine i za njih odgovarajuće, željene izlazne veličine [1].

Postupak učenja se za svaki korak učenja odvija u dvije faze: unaprijedna i povratna faza. Sama promjena parametara se isto može odvijati na dva načina: po skupu (eng. batch) ili po uzorku (eng. pattern). Ukoliko se učenje odvija po skupu nakon svakog ulazno-izlaznog para vrijednost pogreške se sprema i dodaje prethodnoj pogrešci u tom koraku učenja. Nakon što prođu svi ulazno-izlazni parovi i sve vrijednosti grešaka se zbroje, potom se mijenjaju parametri učenja. Tim načinom učenja promjena parametara se odvija nekom srednjom pogreškom učenja. Za razliku od učenja po skupu, učenje po uzorku mijenja parametre učenja nakon svakog ulazno-izlaznog para, tj. nakon što prođe prvi ulazno-izlazni par izračuna se greška te se odmah pristupa mijenjaju parametara po definiranom algoritmu učenja [1]. U ovom radu će se učenje mreže odvijati po uzorku.

2.3.1. Unaprijedna faza učenja statičke neuronske mreže

U unaprijednoj fazi učenja iz datoteke ulaznih veličina uzimaju se vrijednosti svih ulaza mreže Z te se na temelju tih ulaza izračunava izlaz mreže O . Kako bi to bilo moguće potrebno je prvo odrediti iznose težinskih koeficijenata. Najčešće se te vrijednosti određuju generiranjem slučajnih vrijednosti brojeva. Raspon u kojem se odabiru slučajne vrijednosti brojeva je obično u istom redu veličine u kojem se nalaze ulazi i izlazi mreže, odnosno ukoliko su ulazi i izlazi normirani na vrijednosti između -1 i 1, preporučljivo je odabrati slučajne vrijednosti težina također u tom rasponu [1]. Međutim, ukoliko u samom početku učenja pojave problemi sa lokalnim minimumima pogreške, potrebno je interval početnih vrijednosti smanjiti za jedan red u odnosu na interval ulaza i izlaza, tj. na -0,1 i 0,1.

Unaprijedna faza učenja mreže će biti prikazana u poglavlju 3 rada, na primjerima neuronske mreže s jednim i s dva sakrivena sloja.

2.3.2. Povratna faza učenja statičke neuronske mreže

Nakon unaprijedne faze učenja, na osnovu ostvarenog i željenog izlaza mreže izračunava se pogreška učenja. Na temelju pogreške učenja vrši se korekcija vrijednosti težinskih koeficijenata veza između slojeva. Postupak se ponavlja za svaki ulazno-izlazni par podataka u zapisniku za učenje sve dok se ne postigne pogreška manja ili jednaka dozvoljenoj pogrešci, koju ovisno o zadatku zadaje učitelj [1].

Uobičajena statistička metoda regresijske analize, suma kvadrata pogreške kao mjera odstupanja izlaza mreže od željene vrijednosti izlaza najčešće je korištena funkcija cilja:

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - O_n)^2 \quad (2.10)$$

gdje je N broj elemenata u skupu za učenje, odnosno broj podataka učenja iz zapisnika učenja. O_n je dobiveni izlaz mreže, a d_n željeni izlaz mreže. Iz gornje jednadžbe slijedi da je postupak podešavanja težinskih koeficijenata minimizira funkciju cilja. Odabranom funkcijom cilja se vrši promjena koeficijenata težina primjenom nekog od algoritma nelinearnog optimiranja. Forma promjene parametara učenja težinskih koeficijenata ϑ dana je izrazom:

$$\vartheta(n+1) = \vartheta(n) + \Delta\vartheta(n) \quad (2.11)$$

gdje je n trenutni korak učenja, $\Delta\vartheta(n)$ veličina promjene parametara učenja, ϑ predstavlja težine pojedinog sloja mreže, a $\vartheta(n+1)$ je nova vrijednost parametra učenja. Grešku $E(\vartheta)$ moguće je u okolišu točke ϑ aproksimirati s prva dva člana Taylorovog reda:

$$E(\vartheta + \Delta\vartheta) \approx E(\vartheta) + \Delta E(\vartheta) \quad (2.12)$$

$$\Delta E(\vartheta) = \Delta\vartheta^T \nabla E(\vartheta) \quad (2.13)$$

$$\nabla E(\vartheta) = \frac{\partial E(\vartheta)}{\partial \vartheta} \quad (2.14)$$

Izraz (2.14) naziva se gradijentom pogreške. Kako bi se pogreška smanjivala najvećim mogućim iznosom, potrebno je odrediti $\Delta\vartheta$ za koji promjena pogreške učenja $\Delta E(\vartheta)$ poprima najveći negativni iznos, a to se ostvaruje uz uvjet:

$$\Delta\vartheta = -\eta \nabla E(\vartheta) \quad (2.15)$$

gdje je η mjera te promjene, a naziva se koeficijent brzine učenja, čiji iznos određuje učitelj. Izraz (2.15) osigurava promjenu težinskih koeficijenata u smjeru najvećeg mogućeg pada ukupne pogreške. Kombinacijom izraza (2.15) i (2.11) dobiva se:

$$\vartheta(n+1) = \vartheta(n) - \eta \nabla E(\vartheta(n)). \quad (2.16)$$

Izraz (2.16) poznat je u literaturi o nelinearnom optimiranju pod nazivom algoritam najstrmijeg pada, a prema literaturi o neuronskim mrežama pod nazivom algoritam povratnog prostiranja pogreške i predstavlja najpoznatiji i najčešće primjenjivani način promjene parametara učenja [1]. Najveći nedostatak tog algoritma je veliki broj potrebnih iteracija. Kako bi se ubrzao proces učenja, tj. kako bi se smanjio potreban broj koraka učenja uz istu dozvoljenu pogrešku učenja, algoritam se modificira ugradnjom momentuma (zamaha) prvog i drugog reda u izraz (2.15):

$$\Delta\vartheta(n) = -\eta \nabla E(\vartheta(n)) + \alpha \Delta\vartheta(n-1) + \beta \Delta\vartheta(n-2) \quad (2.17)$$

gdje su:

- (n) – trenutna promjena parametara učenja
- $(n-1)$ – prošla promjena parametara učenja
- $(n-2)$ – pretprošla promjena parametara učenja
- α – vrijednost koeficijenta momentuma prvog reda. Tu vrijednost zadaje učitelj, a obično se bira u intervalu od 0,1 do 0,9
- β – vrijednost koeficijenta momentuma drugog reda. Ta vrijednost se računa u odnosu na vrijednost koeficijenta momentuma prvog reda prema sljedećem izrazu:

$$\beta = (1 - \alpha) / 3. \quad (2.18)$$

Konačni izraz za promjenu parametara učenja glasi:

$$\vartheta(n+1) = \vartheta(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta\vartheta(n-1) + \beta \Delta\vartheta(n-2). \quad (2.19)$$

Ugradnjom momentuma u izraz (2.15) brzina algoritma se u odnosu na klasični algoritam povratnog prostiranja pogreške povećava i do nekoliko puta, ali zato ne garantira konvergenciju.

3. MATEMATIČKI IZVOD ZADANIH MODELA NEURONSKIH MREŽA

Slijedi prikaz modela mreža s jednim sakrivenim slojem i s dva sakrivena sloja, te za svaku mrežu zasebno izračun unaprijedne i povratne faze učenja za zadane ulazno-izlazne parove podataka.

3.1. Neuronska mreža s jednim sakrivenim slojem

Na slici 2.3. prikazan je model statičke unaprijedne neuronske mreže s jednim sakrivenim slojem. Prema strukturi prikazanoj na slici, i s tako definiranim veličinama biti će opisane faze učenja te mreže.

3.1.1. Unaprijedna faza učenja neuronske mreže s jednim sakrivenim slojem

Unaprijedna faza učenja neuronske mreže s jednim sakrivenim slojem odvija se od ulaznog prema izlaznom sloju mreže.

3.1.1.1. Sakriveni sloj

Funkcija sume net neurona sakrivenog sloja H dobiva za prvi indeks oznaku sloja, a za svaki j – ti neuron dobiva drugi indeks i računa se na sljedeći način:

$$net_{Hj} = \sum_{i=1}^I v_{ji} Z_i \quad , \quad j=1,2,\dots,J-1 \quad , \quad i=1,2,\dots,I \quad (3.1)$$

gdje je I broj ulaznih neurona plus 1 (BIAS), J broj neurona u sakrivenom sloju uvećan za jedan (BIAS), a v_{ji} težinski koeficijenti preko kojih su povezani ulazi mreže s ulazima neurona sakrivenog sloja. Ako se za aktivacijsku funkciju sakrivenog sloja odabere nelinearna bipolarna sigmoidalna funkcija onda se izlazi neurona sakrivenog sloja računaju prema izrazu:

$$y_j = \frac{2}{1+e^{-net_{Hj}}} - 1, \quad (3.2)$$

a ako se odabere sinusna funkcija kao aktivacijska funkcija sakrivenog sloja, izlazi neurona se računaju prema izrazu:

$$y_j = \sin (net_{Hj}) \quad (3.3)$$

gdje je $j = 1, 2, \dots, J-1$, a $y_J = 1$ (BIAS). Izračunate vrijednosti izlaza neurona sakrivenog sloja povezane su sa ulazima neurona izlaznog sloja preko težinskih koeficijenata w_{kj} .

3.1.1.2. Izlazni sloj

Funkcija sume net neurona izlaznog sloja O dobiva prvi indeks pripadnog sloja, te drugi pripadni indeks svakom neuronu izlaznog sloja, a računa se prema izrazu:

$$net_{Ok} = \sum_{j=1}^J w_{kj} y_j, \quad k=1,2,\dots,K \quad (3.4)$$

gdje je K broj neurona izlaznog sloja, tj. broj izlaza mreže.

Za aktivacijsku funkciju izlaznog sloja se koristi linearna aktivacijska funkcija, kako bi se ostvarile vrijednosti izlaza mreže veće od 1:

$$O_k = K_p \cdot net_{Ok}, \quad k=1,2,\dots,K \quad (3.5)$$

gdje je K_p nagib linearne aktivacijske funkcije, a za njegov iznos će se u nastavku koristiti 1. Moguće je koristiti neke druge aktivacijske funkcije, primjerice sigmoidalnu aktivacijsku funkciju, ali ako se koristi funkcija sa zasićenjem na izlazu nije moguće ostvariti izlaze iz neuronske mreže veće od 1 i manje od -1, te mreža ne bi dobro radila pošto će vrijednosti ulazno-izlaznih parova podataka za učenje biti izvan tih granica.

3.1.2. Povratna faza učenja neuronske mreže s jednim sakrivenim slojem

U poglavlju 2.3.2. izveden je algoritam prema kojem se odvija promjena parametara učenja neuronske mreže (u slučaju statičke neuronske mreže to su jedino težinski koeficijenti veza među slojevima neurona). Povratna faza se odvija od izlaznog sloja prema ulaznom sloju mreže.

3.1.2.1. Promjena težina izlaznog sloja

Promjena težinskih faktora veza između izlaznog i sakrivenog sloja odvija se prema izrazu (2.19) na slijedeći način:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta w_{kj}(n-1) + \beta \Delta w_{kj}(n-2). \quad (3.6)$$

Gradijent pogreške za težine w_{kj} računa se prema izrazu (2.14):

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kj}}. \quad (3.7)$$

Prema izrazu (3.7) osnovni zadatak postupka učenja je određivanje pripadajućeg gradijenta pogreške. Gornji izraz se jednostavno rješava primjenom uzastopnih parcijalnih derivacija:

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial w_{kj}}. \quad (3.8)$$

Prema izrazu (2.10), prvi član izraza (3.8) iznosi:

$$\frac{\partial E(n)}{\partial O_k} = - (d_k - O_k). \quad (3.9)$$

Drugi član izraza (3.8) dobiva se iz izraza (3.5):

$$\frac{\partial O_k}{\partial net_{Ok}} = 1. \quad (3.10)$$

Karakteristična vrijednost algoritma povratnog prostiranja greške δ , koja se računa prvo za izlazni sloj je definirana kao:

$$\delta = - \frac{\partial E(n)}{\partial net}. \quad (3.11)$$

Na temelju izraza (3.8), (3.9), (3.10) i (3.11) vrijednost δ izlaznog sloja neurona u ovo slučaju ima oblik:

$$\delta_{Ok} = d_k - O_k. \quad (3.12)$$

Posljednji član izraza (3.8) određuje se prema izrazu (3.4):

$$\frac{\partial net_{Ok}}{\partial w_{kj}} = y_j. \quad (3.13)$$

Uvrštavanjem izraza (3.9), (3.10), (3.12) i (3.13) u izraz (3.8) dobiva se:

$$\nabla E(\vartheta) = \frac{\partial E(n)}{\partial w_{kj}} = \delta_{Ok} y_j. \quad (3.14)$$

Ako se izraz (3.14) uvrsti u izraz (3.6) dobiva se konačni algoritam promjene težina izlaznog sloja:

$$w_{kj}(n+1) = w_{kj}(n) + \eta \delta_{Ok} y_j + \alpha \Delta w_{kj}(n-1) + \beta \Delta w_{kj}(n-2). \quad (3.15)$$

3.1.2.2. Promjena težina sakrivenog sloja

Nakon promjene svih težina w_{kj} izlaznog sloja, slijedi promjena težina sakrivenog sloja v_{ji} . Promjena težinskih faktora veza između sakrivenog i ulaznog sloja odvija se prema izrazu (2.19) na slijedeći način:

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta v_{ji}(n-1) + \beta \Delta v_{ji}(n-2). \quad (3.16)$$

Za težine sakrivenog sloja, kao i za težine izlaznog sloja, jedini je problem izračunati gradijent greške $\nabla E(n)$. Uzastopnim parcijalnim deriviranjem prema slici 2.3. proizlazi:

$$\frac{\partial E(n)}{\partial v_{ji}} = \frac{\partial E(n)}{\partial y_j} \frac{\partial y_j}{\partial net_{Hj}} \frac{\partial net_{Hj}}{\partial v_{ji}}. \quad (3.17)$$

Slika 2.3. pokazuje da na promjenu svake težine sakrivenog sloja utječu svi neuroni izlaznog sloja, pa prvi član s desne strane u izrazu (3.17) poprima oblik:

$$\frac{\partial E(n)}{\partial y_j} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial y_j}. \quad (3.18)$$

Pri tome je:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) = \delta_{Ok}, \quad k=1,2,\dots,K, \quad (3.19)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1, \quad k=1,2,\dots,K, \quad (3.20)$$

$$\frac{\partial net_{Ok}}{\partial y_j} = w_{kj}, \quad k=1,2,\dots,K, \quad j=1,2,\dots,J-1. \quad (3.21)$$

Uvrštavanjem izraza (3.19), (3.20) i (3.21) u izraz (3.18) dobiva se:

$$\frac{\partial E(n)}{\partial y_j} = - \sum_{k=1}^K \delta_{ok} w_{kj}. \quad (3.22)$$

Drugi član s desne strane izraza (3.17) određuje se prema izrazu (3.2) ili (3.3) ovisno koja aktivacijska funkcija se koristi. Ako se koristi sigmoidalna aktivacijska funkcija onda je:

$$\frac{\partial y_j}{\partial net_{Hj}} = \frac{1}{2} (1 - y_j^2), \quad (3.23)$$

a ako se koristi sinusna aktivacijska funkcija taj izraz se mijenja u:

$$\frac{\partial y_j}{\partial net_{Hj}} = \cos (net_{Hj}). \quad (3.24)$$

Posljednji član s desne strane izraza (3.17) prema izrazu (3.2) iznosi:

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i. \quad (3.25)$$

Uvrštavanjem izraza (3.23), (3.24) i (3.25) u izraz (3.17) dobiva se konačno rješenje gradijenta pogreške:

$$\nabla E(n) = - \frac{1}{2} (1 - y_j^2) Z_i \left(\sum_{k=1}^K \delta_{ok} w_{kj} \right), \quad (3.26)$$

za sigmoidalnu aktivacijsku funkciju, odnosno:

$$\nabla E(n) = - \cos (net_{Hj}) Z_i \left(\sum_{k=1}^K \delta_{ok} w_{kj} \right) \quad (3.27)$$

za sinusnu aktivacijsku funkciju. Uvrštavanjem izraza (3.26), odnosno (3.27) u izraz (3.16) dobiva se konačan algoritam promjene težina sakrivenog sloja:

$$v_{ji}(n+1) = v_{ji}(n) + \eta \frac{1}{2} (1 - y_j^2) Z_i \left(\sum_{k=1}^K \delta_{ok} w_{kj} \right) + \alpha \Delta v_{ji}(n-1) + \beta \Delta v_{ji}(n-2) \quad (3.28)$$

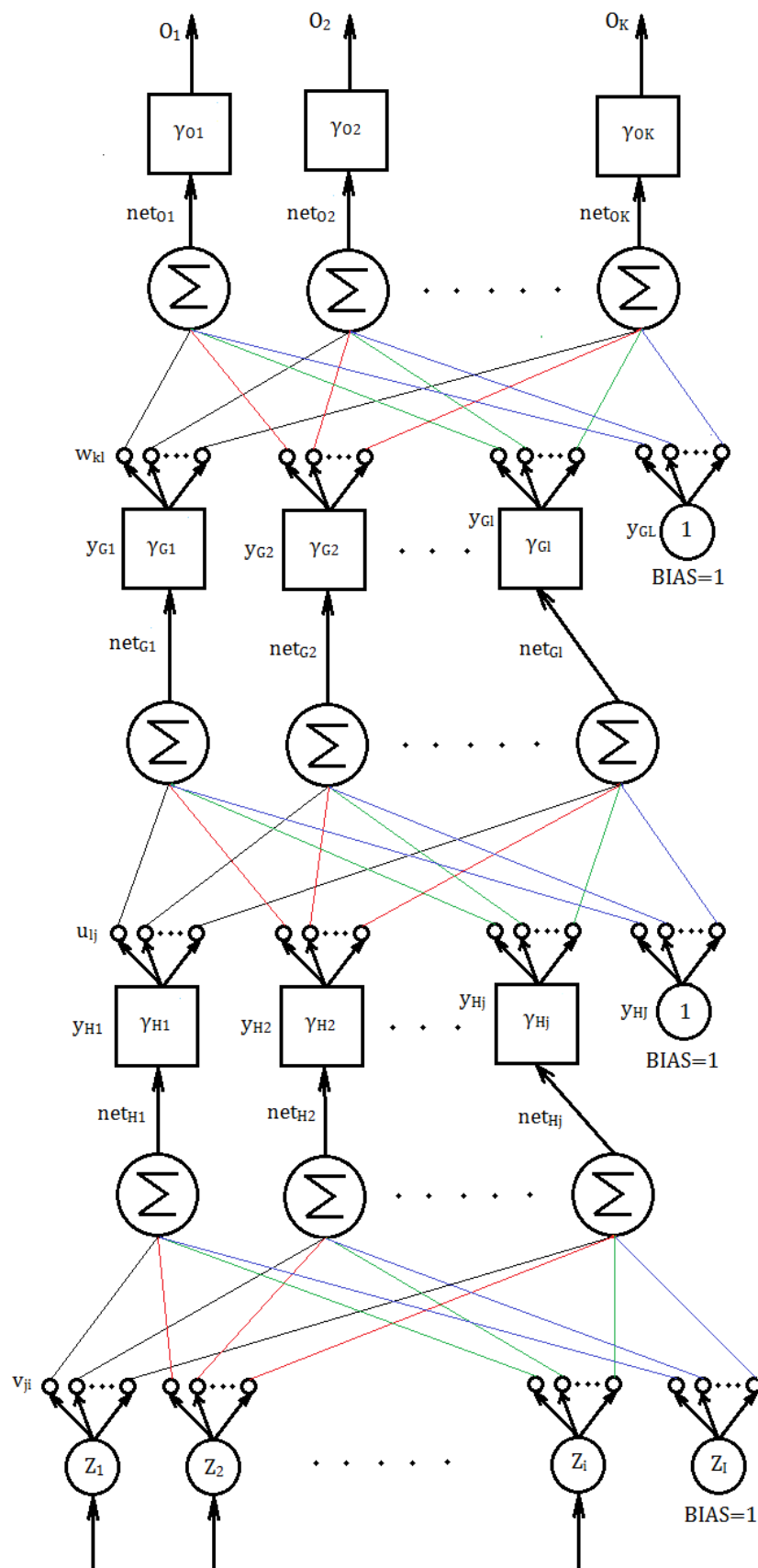
za sigmoidalnu aktivacijsku funkciju, te:

$$v_{ji}(n+1) = v_{ji}(n) + \eta \cos (net_{Hj}) Z_i \left(\sum_{k=1}^K \delta_{ok} w_{kj} \right) + \alpha \Delta v_{ji}(n-1) + \beta \Delta v_{ji}(n-2) \quad (3.29)$$

za sinusnu aktivacijsku funkciju.

3.2. Neuronska mreža s dva sakrivena sloja

Numerički postupak za neuronsku mrežu s dva sakrivena sloja je potpuno analogan postupku za mrežu s jednim sakrivenim slojem, samo se izrazi dodatno proširuju. Na slici 3.1. prikazana je struktura mreže s dva sakrivena sloja. Sve veličine su definirane na slici, te će za tako definiranu strukturu mreže biti izvedene unaprijedna i povratna faza učenja mreže.



Slika 3.1. Struktura statičke unaprijedne neuronske mreže s dva sakrivena sloja

3.2.1. Unaprijedna faza učenja neuronske mreže s dva sakrivena sloja

Unaprijedna faza se isto kao i kod mreže s jednim sakrivenim slojem odvija od ulaznog prema izlaznom sloju mreže.

3.2.1.1. Prvi sakriveni sloj

Funkcija sume net neurona prvog sakrivenog sloja H dobiva za prvi indeks oznaku sloja, a za svaki j – ti neuron dobiva drugi indeks i računa se na sljedeći način:

$$net_{Hj} = \sum_{i=1}^I v_{ji} Z_i \quad , \quad j=1,2,\dots,J-1 \quad , \quad i=1,2,\dots,I \quad (3.30)$$

gdje je I broj ulaznih neurona plus 1 (BIAS), J broj neurona u sakrivenom sloju uvećan za jedan (BIAS), a v_{ji} težinski koeficijenti preko kojih su povezani ulazi mreže s ulazima neurona prvog sakrivenog sloja. Ako se za aktivacijsku funkciju sakrivenog sloja odabere nelinearna bipolarna sigmoidalna funkcija onda se izlazi neurona sakrivenog sloja računaju prema izrazu:

$$y_{Hj} = \frac{2}{1+e^{-net_{Hj}}} - 1, \quad (3.31)$$

a ako se odabere sinusna funkcija kao aktivacijska funkcija sakrivenog sloja, izlazi neurona se računaju prema izrazu:

$$y_{Hj} = \sin (net_{Hj}) \quad (3.32)$$

gdje je $j = 1, 2, \dots, J-1$, a $y_{Hj} = 1$ (BIAS). Izračunate vrijednosti izlaza neurona prvog sakrivenog sloja povezane su sa ulazima neurona drugog sakrivenog sloja preko težinskih koeficijenata u_{lj} .

3.2.1.2. Drugi sakriveni sloj

Funkcija sume net neurona drugog sakrivenog sloja G dobiva za prvi indeks oznaku sloja, a za svaki l – ti neuron dobiva drugi indeks i računa se na sljedeći način:

$$net_{Gl} = \sum_{j=1}^J u_{lj} y_{Hj} \quad , \quad l=1,2,\dots,L-1 \quad , \quad j=1,2,\dots,J \quad (3.33)$$

gdje je J broj neurona prvog sakrivenog sloja plus 1 (BIAS), a L broj neurona drugog sakrivenog sloja uvećan za jedan (BIAS). Ako se za aktivacijsku funkciju drugog sakrivenog

sloja odabere nelinearna bipolarna sigmoidalna funkcija onda se izlazi neurona sakrivenog sloja računaju prema izrazu:

$$y_{Gl} = \frac{2}{1+e^{-net_{Gl}}} - 1, \quad (3.34)$$

a ako se odabere sinusna funkcija kao aktivacijska funkcija sakrivenog sloja, izlazi neurona se računaju prema izrazu:

$$y_{Gl} = \sin (net_{Gl}) \quad (3.35)$$

gdje je $l = 1, 2, \dots, L-1$, a $y_{GL} = 1$ (BIAS). Izračunate vrijednosti izlaza neurona drugog sakrivenog sloja povezane su sa ulazima neurona izlaznog sloja preko težinskih koeficijenata w_{kl} .

3.2.1.3. Izlazni sloj

Funkcija sume net neurona izlaznog sloja O dobiva prvi indeks pripadnog sloja, te drugi pripadni indeks svakom neuronu izlaznog sloja, a računa se prema izrazu:

$$net_{Ok} = \sum_{l=1}^L w_{kl} y_{Gl}, \quad k=1,2,\dots,K \quad (3.36)$$

gdje je K broj neurona izlaznog sloja, tj. broj izlaza mreže.

Za aktivacijsku funkciju izlaznog sloja se koristi linearna aktivacijska funkcija, iz istih razloga kao i kod mreže s jednim sakrivenim slojem:

$$O_k = K_p \cdot net_{Ok}, \quad k=1,2,\dots,K \quad (3.37)$$

gdje je K_p nagib linearne aktivacijske funkcije izlaznog sloja i njegov iznos je 1.

3.2.2. Povratna faza učenja neuronske mreže s dva sakrivena sloja

Povratna faza učenja mreže s dva sakrivena sloja se odvija od izlaznog prema ulaznom sloju neuronske mreže, isto kao i kod mreže s jednim sakrivenim slojem. Algoritam izveden u poglavlju 2.3.2. isto se koristi za mrežu s dva sakrivena sloja.

3.2.2.1. Promjena težina izlaznog sloja

Promjena težinskih faktora veza između izlaznog i drugog sakrivenog sloja odvija se prema izrazu (2.19) na slijedeći način:

$$w_{kl}(n+1) = w_{kl}(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta w_{kl}(n-1) + \beta \Delta w_{kl}(n-2). \quad (3.38)$$

Gradijent pogreške za težine w_{kl} računa se prema izrazu (2.14):

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kl}}. \quad (3.39)$$

Izraz (3.39) se jednostavno rješava primjenom uzastopnih parcijalnih derivacija:

$$\frac{\partial E(n)}{\partial w_{kl}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial w_{kl}}. \quad (3.40)$$

Prema izrazu (2.10), prvi član izraza (3.40) iznosi:

$$\frac{\partial E(n)}{\partial O_k} = - (d_k - O_k). \quad (3.41)$$

Drugi član izraza (3.40) dobiva se iz izraza (3.37):

$$\frac{\partial O_k}{\partial net_{Ok}} = 1. \quad (3.42)$$

Posljednji član izraza (3.40) određuje se prema izrazu (3.36):

$$\frac{\partial net_{Ok}}{\partial w_{kl}} = y_{Gl}. \quad (3.43)$$

Uvrštavanjem izraza (3.41), (3.42) i (3.43) u izraz (3.40), te uz pomoć izraza (3.11) dobiva se:

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kl}} = \delta_{Ok} y_{Gl}. \quad (3.44)$$

Ako se izraz (3.44) uvrsti u izraz (3.38) dobiva se konačni algoritam promjene težina izlaznog sloja:

$$w_{kl}(n+1) = w_{kl}(n) + \eta \delta_{Ok} y_{Gl} + \alpha \Delta w_{kl}(n-1) + \beta \Delta w_{kl}(n-2). \quad (3.45)$$

3.2.2.2. Promjena težina drugog sakrivenog sloja

Nakon promjene svih težina w_{kl} izlaznog sloja, slijedi promjena težina drugog sakrivenog sloja u_{lj} . Promjena težinskih faktora veza između prvog sakrivenog i drugog sakrivenog sloja odvija se prema izrazu (2.19) na slijedeći način:

$$u_{lj}(n+1) = u_{lj}(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta u_{lj}(n-1) + \beta \Delta u_{lj}(n-2). \quad (3.46)$$

Uzastopnim parcijalnim deriviranjem prema slici 6 proizlazi da je gradijent pogreške drugog sakrivenog sloja:

$$\frac{\partial E(n)}{\partial u_{lj}} = \frac{\partial E(n)}{\partial y_{Gl}} \frac{\partial y_{Gl}}{\partial net_{Gl}} \frac{\partial net_{Gl}}{\partial u_{lj}}. \quad (3.47)$$

Slika 3.1. pokazuje da na promjenu svake težine drugog sakrivenog sloja utječu svi neuroni izlaznog sloja, pa prvi član s desne strane u izrazu (3.47) poprima oblik:

$$\frac{\partial E(n)}{\partial y_{Gl}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial y_{Gl}}. \quad (3.48)$$

Pri tome je:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) = \delta_{Ok}, \quad k = 1, 2, \dots, K, \quad (3.49)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1, \quad k = 1, 2, \dots, K, \quad (3.50)$$

$$\frac{\partial net_{Ok}}{\partial y_{Hj}} = w_{kl}, \quad k = 1, 2, \dots, K, \quad l = 1, 2, \dots, L-1. \quad (3.51)$$

Uvrštavanjem izraza (3.49), (3.50) i (3.51) u izraz (3.48) dobiva se:

$$\frac{\partial E(n)}{\partial y_{Hj}} = - \sum_{k=1}^K \delta_{Ok} w_{kl}. \quad (3.52)$$

Drugi član s desne strane izraza (3.47) određuje se prema izrazu (3.34) ili (3.35) ovisno koja aktivacijska funkcija se koristi. Ako se koristi sigmoidalna aktivacijska funkcija onda je:

$$\frac{\partial y_{Gl}}{\partial net_{Gl}} = \frac{1}{2} (1 - y_{Gl}^2), \quad (3.53)$$

a ako se koristi sinusna aktivacijska funkcija taj izraz glasi:

$$\frac{\partial y_{Gl}}{\partial net_{Gl}} = \cos (net_{Gl}) . \quad (3.54)$$

Posljednji član s desne strane izraza (3.47) prema izrazu (3.33) iznosi:

$$\frac{\partial net_{Gl}}{\partial u_{lj}} = y_{Hj} \quad (3.55)$$

Uvrštavanjem izraza (3.52), (3.53), odnosno (3.54) i (3.55) u izraz (3.47) dobiva se konačno rješenje gradijenta pogreške drugog sakrivenog sloja:

$$\nabla E(n) = -\frac{1}{2} (1 - y_{Gl}^2) y_{Hj} (\sum_{k=1}^K \delta_{Ok} w_{kl}) \quad (3.56)$$

za sigmoidalnu aktivacijsku funkciju, odnosno:

$$\nabla E(n) = -\cos (net_{Gl}) y_{Hj} (\sum_{k=1}^K \delta_{Ok} w_{kl}) \quad (3.57)$$

za sinusnu aktivacijsku funkciju. Uvrštavanjem izraza (3.56), odnosno (3.57) u izraz (3.46) dobiva se konačan algoritam promjene težina drugog sakrivenog sloja:

$$u_{lj}(n+1) = u_{lj}(n) + \eta \frac{1}{2} (1 - y_{Gl}^2) y_{Hj} (\sum_{k=1}^K \delta_{Ok} w_{kl}) + \alpha \Delta u_{lj}(n-1) + \beta \Delta u_{lj}(n-2) \quad (3.58)$$

za sigmoidalnu aktivacijsku funkciju, te:

$$u_{lj}(n+1) = u_{lj}(n) + \eta \cos (net_{Gl}) y_{Hj} (\sum_{k=1}^K \delta_{Ok} w_{kl}) + \alpha \Delta u_{lj}(n-1) + \beta \Delta u_{lj}(n-2) \quad (3.59)$$

za sinusnu aktivacijsku funkciju.

3.2.2.3. Promjena težina prvog sakrivenog sloja

Nakon promjene svih težina u_{lj} drugog sakrivenog sloja, slijedi promjena težina prvog sakrivenog sloja v_{ji} . Promjena težinskih faktora veza između prvog sakrivenog i ulaznog sloja odvija se prema izrazu (2.19) na slijedeći način:

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta v_{ji}(n-1) + \beta \Delta v_{ji}(n-2). \quad (3.60)$$

Uzastopnim parcijalnim deriviranjem prema slici 3.1. proizlazi da je gradijent greške prvog sakrivenog sloja:

$$\frac{\partial E(n)}{\partial v_{ji}} = \frac{\partial E(n)}{\partial y_{Hj}} \frac{\partial y_{Hj}}{\partial net_{Hj}} \frac{\partial net_{Hj}}{\partial v_{ji}}. \quad (3.61)$$

Slika 3.1. pokazuje da na promjenu svake težine prvog sakrivenog sloja utječu svi neuroni drugog sakrivenog sloja, pa prvi član u izrazu (3.61) poprima oblik:

$$\frac{\partial E(n)}{\partial y_{Hj}} = \frac{\partial E(n)}{\partial y_{Gl}} \frac{\partial y_{Gl}}{\partial net_{Gl}} \frac{\partial net_{Gl}}{\partial y_{Hj}}. \quad (3.62)$$

Prvi i drugi član s desne strane izraza (3.62) su već definirani u poglavlju 3.2.2.2. te neće biti izvođeni ovdje. Treći član prema izrazu (3.33) iznosi:

$$\frac{\partial net_{Gl}}{\partial y_{Hj}} = u_{lj}. \quad (3.63)$$

Uvrštavanjem izraza (3.52), (3.53), odnosno (3.54) i (3.63) u izraz (3.62) dobiva se:

$$\frac{\partial E(n)}{\partial y_{Hj}} = - \sum_{l=1}^L \left(\frac{1}{2} (1 - y_{Gl}^2) u_{lj} (\sum_{k=1}^K \delta_{Ok} w_{kl}) \right) \quad (3.64)$$

za sigmoidalnu aktivacijsku funkciju i:

$$\frac{\partial E(n)}{\partial y_{Hj}} = - \sum_{l=1}^L (\cos(net_{Gl}) u_{lj} (\sum_{k=1}^K \delta_{Ok} w_{kl})) \quad (3.65)$$

za sinusnu aktivacijsku funkciju.

Drugi član s desne strane izraza (3.61) određuje se prema izrazu (3.31) ili (3.32) ovisno koja aktivacijska funkcija se koristi. Ako se koristi sigmoidalna aktivacijska funkcija onda je:

$$\frac{\partial y_{Hj}}{\partial net_{Hj}} = \frac{1}{2} (1 - y_{Hj}^2), \quad (3.66)$$

a ako se koristi sinusna aktivacijska funkcija taj izraz glasi:

$$\frac{\partial y_{Hj}}{\partial net_{Hj}} = \cos(net_{Hj}). \quad (3.67)$$

Posljednji član s desne strane izraza (3.61) prema izrazu (3.30) iznosi:

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i. \quad (3.68)$$

Uvrštavanjem izraza (3.65), (3.66), odnosno (3.67) i (3.68) u izraz (3.61) dobiva se konačno rješenje gradijenta pogreške prvog sakrivenog sloja:

$$\nabla E(n) = -\frac{1}{2} (1 - y_{Hj}^2) Z_i \left(\sum_{l=1}^L \left(\frac{1}{2} (1 - y_{Gl}^2) u_{lj} \left(\sum_{k=1}^K \delta_{Ok} w_{kl} \right) \right) \right) \quad (3.69)$$

za sigmoidalnu aktivacijsku funkciju, odnosno:

$$\nabla E(n) = -\cos (net_{Hj}) Z_i \left(\sum_{l=1}^L \left(\cos (net_{Gl}) u_{lj} \left(\sum_{k=1}^K \delta_{Ok} w_{kl} \right) \right) \right) \quad (3.70)$$

za sinusnu aktivacijsku funkciju. Uvrštavanjem izraza (3.69), odnosno (3.70) u izraz (3.60) dobiva se konačan algoritam promjene težina prvog sakrivenog sloja:

$$\begin{aligned} v_{ji}(n+1) = v_{ji}(n) + \eta \frac{1}{2} (1 - y_{Hj}^2) Z_i \left(\sum_{l=1}^L \left(\frac{1}{2} (1 - y_{Gl}^2) u_{lj} \left(\sum_{k=1}^K \delta_{Ok} w_{kl} \right) \right) \right) \\ + \alpha \Delta v_{ji}(n-1) + \beta \Delta v_{ji}(n-2) \end{aligned} \quad (3.71)$$

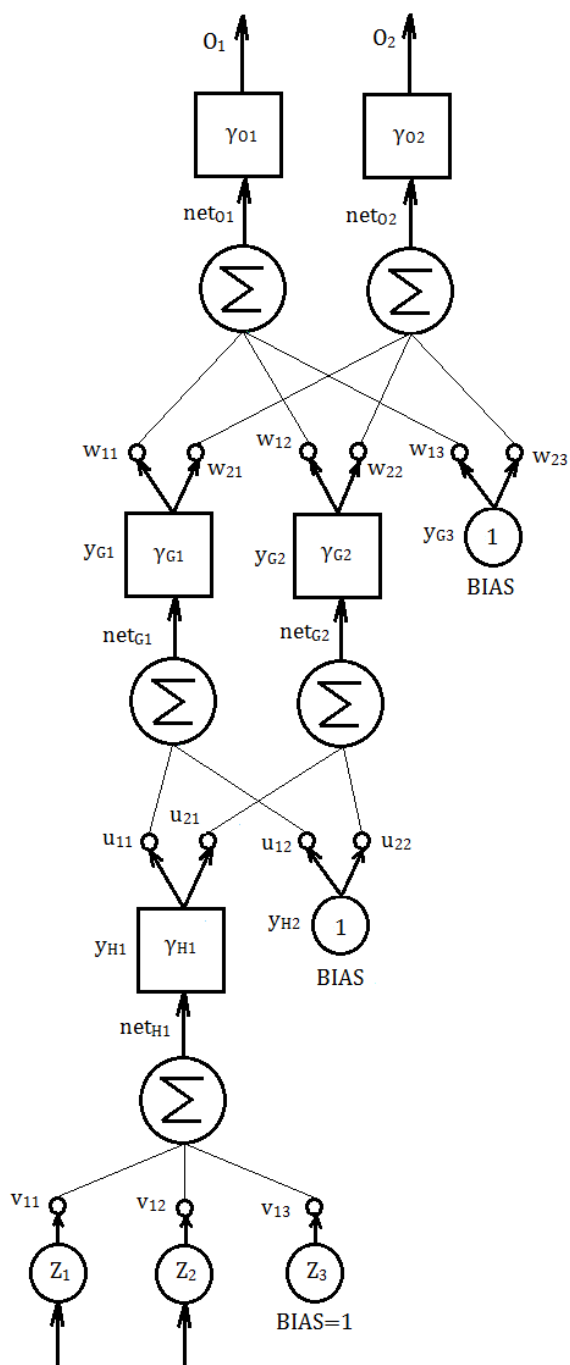
za sigmoidalnu aktivacijsku funkciju, te:

$$\begin{aligned} v_{ji}(n+1) = v_{ji}(n) + \eta \cos (net_{Hj}) Z_i \left(\sum_{l=1}^L \left(\cos (net_{Gl}) u_{lj} \left(\sum_{k=1}^K \delta_{Ok} w_{kl} \right) \right) \right) \\ + \alpha \Delta v_{ji}(n-1) + \beta \Delta v_{ji}(n-2) \end{aligned} \quad (3.72)$$

za sinusnu aktivacijsku funkciju.

3.2.3. Primjer neuronske mreže s dva sakrivena sloja

Slijedi prikaz izvoda matematičkog modela neuronske mreže na primjeru s parametrima: $I = 3$, $J = 2$, $L = 3$, $K = 2$ i sinusnom aktivacijskom funkcijom. Na slici 3.2. je prikazana struktura te mreže.



Slika 3.2. Struktura neuronske mreže za zadani primjer

Raspisani izrazi za zadani primjer:

$$net_{H1} = Z_1 v_{11} + Z_2 v_{12} + Z_3 v_{13}$$

$$y_{H1} = \sin(net_{H1})$$

$$y_{H2} = 1$$

$$net_{G1} = u_{11} y_{H1} + u_{12} y_{H2}$$

$$net_{G2} = u_{21} y_{H1} + u_{22} y_{H2}$$

$$y_{G1} = \sin(net_{G1})$$

$$y_{G2} = \sin(net_{G1})$$

$$y_{G3} = 1$$

$$net_{O1} = w_{11} y_{G1} + w_{12} y_{G2} + w_{13} y_{G3}$$

$$net_{O2} = w_{21} y_{G1} + w_{22} y_{G2} + w_{23} y_{G3}$$

$$\frac{\partial E(n)}{\partial w_{11}} = \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial w_{11}} = - (d_1 - O_1) y_{G1} = - \delta_{O1} y_{G1}$$

$$\frac{\partial E(n)}{\partial w_{21}} = \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial w_{21}} = - (d_2 - O_2) y_{G1} = - \delta_{O2} y_{G1}$$

$$\frac{\partial E(n)}{\partial w_{12}} = \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial w_{12}} = - (d_1 - O_1) y_{G2} = - \delta_{O1} y_{G2}$$

$$\frac{\partial E(n)}{\partial w_{22}} = \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial w_{22}} = - (d_2 - O_2) y_{G2} = - \delta_{O2} y_{G2}$$

$$\frac{\partial E(n)}{\partial w_{13}} = \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial w_{13}} = - (d_1 - O_1) y_{G3} = - \delta_{O1} y_{G3}$$

$$\frac{\partial E(n)}{\partial w_{23}} = \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial w_{23}} = - (d_2 - O_2) y_{G3} = - \delta_{O2} y_{G3}$$

$$\frac{\partial E(n)}{\partial u_{11}} = \frac{\partial E(n)}{\partial y_{G1}} \frac{\partial y_{G1}}{\partial net_{G1}} \frac{\partial net_{G1}}{\partial u_{11}}$$

$$\frac{\partial E(n)}{\partial y_{G1}} = \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial y_{G1}} + \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial y_{G1}} =$$

$$= - (d_1 - O_1) w_{11} - (d_2 - O_2) w_{21} = - (\delta_{O1} w_{11} + \delta_{O2} w_{21})$$

$$\frac{\partial E(n)}{\partial u_{11}} = - (\delta_{O1} w_{11} + \delta_{O2} w_{21}) \cos(net_{G1}) y_{H1}$$

$$\frac{\partial E(n)}{\partial u_{21}} = \frac{\partial E(n)}{\partial y_{G2}} \frac{\partial y_{G2}}{\partial net_{G2}} \frac{\partial net_{G2}}{\partial u_{21}}$$

$$\begin{aligned} \frac{\partial E(n)}{\partial y_{G2}} &= \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial y_{G2}} + \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial y_{G2}} = \\ &= - (d_1 - O_1) w_{12} - (d_2 - O_2) w_{22} = - (\delta_{O1} w_{12} + \delta_{O2} w_{22}) \end{aligned}$$

$$\frac{\partial E(n)}{\partial u_{21}} = - (\delta_{O1} w_{12} + \delta_{O2} w_{22}) \cos(net_{G2}) y_{H1}$$

$$\frac{\partial E(n)}{\partial v_{11}} = \frac{\partial E(n)}{\partial y_{H1}} \frac{\partial y_{H1}}{\partial net_{H1}} \frac{\partial net_{H1}}{\partial v_{11}}$$

$$\begin{aligned} \frac{\partial E(n)}{\partial y_{H1}} &= \frac{\partial E(n)}{\partial y_{G1}} \frac{\partial y_{G1}}{\partial net_{G1}} \frac{\partial net_{G1}}{\partial y_{H1}} + \frac{\partial E(n)}{\partial y_{G2}} \frac{\partial y_{G2}}{\partial net_{G2}} \frac{\partial net_{G2}}{\partial y_{H1}} + \frac{\partial E(n)}{\partial y_{G3}} \frac{\partial y_{G3}}{\partial net_{G3}} \frac{\partial net_{G3}}{\partial y_{H1}} = \\ &= - (\delta_{O1} w_{11} + \delta_{O2} w_{21}) \cos(net_{G1}) u_{11} - (\delta_{O1} w_{12} + \delta_{O2} w_{22}) \cos(net_{G2}) u_{11} \end{aligned}$$

$$\begin{aligned} \frac{\partial E(n)}{\partial v_{11}} &= - ((\delta_{O1} w_{11} + \delta_{O2} w_{21}) \cos(net_{G1}) u_{11} + \\ &\quad + (\delta_{O1} w_{12} + \delta_{O2} w_{22}) \cos(net_{G2}) u_{11}) \cos(net_{H1}) Z_1 \end{aligned}$$

$$\begin{aligned} \frac{\partial E(n)}{\partial v_{12}} &= \frac{\partial E(n)}{\partial y_{H1}} \frac{\partial y_{H1}}{\partial net_{H1}} \frac{\partial net_{H1}}{\partial v_{12}} = \\ &= - ((\delta_{O1} w_{11} + \delta_{O2} w_{21}) \cos(net_{G1}) u_{11} + \\ &\quad + (\delta_{O1} w_{12} + \delta_{O2} w_{22}) \cos(net_{G2}) u_{11}) \cos(net_{H1}) Z_2 \end{aligned}$$

$$\begin{aligned} \frac{\partial E(n)}{\partial v_{13}} &= \frac{\partial E(n)}{\partial y_{H1}} \frac{\partial y_{H1}}{\partial net_{H1}} \frac{\partial net_{H1}}{\partial v_{13}} = \\ &= - ((\delta_{O1} w_{11} + \delta_{O2} w_{21}) \cos(net_{G1}) u_{11} + \\ &\quad + (\delta_{O1} w_{12} + \delta_{O2} w_{22}) \cos(net_{G2}) u_{11}) \cos(net_{H1}) Z_3 \end{aligned}$$

3.3. Ocjena točnosti algoritma učenja

Kako bi se ocijenila točnost algoritma učenja za određeni zadatak potrebno je definirati mjeru točnosti. Najčešće su u upotrebi tri mjere točnosti učenja [1]:

- Srednja kvadratna pogreška (eng. Mean Square error):

$$MS = \frac{\sum_{n=1}^N (d_n - O_n)^2}{N} \quad (3.73)$$

gdje je N ukupan broj izlaza.

- Korijen srednje kvadratne pogreške (eng. Root Mean Square error):

$$RMS = \sqrt{MS}. \quad (3.74)$$

- Normalizirani korijen srednje kvadratne pogreške (eng. Normalized Root Mean Square error):

$$NRMS = \frac{RMS}{\sigma_{d_n}}, \quad (3.75)$$

gdje je σ_{d_n} :

$$\sigma_{d_n} = \sqrt{\frac{1}{N} \sum_{n=1}^N (d_n - \bar{d})^2}, \quad (3.76)$$

a \bar{d} je:

$$\bar{d} = \frac{1}{N} \sum_{n=1}^N d_n. \quad (3.77)$$

Normalizirani korijen srednje kvadratne pogreške će se koristiti kao mjera točnosti. Prednost NRMS – a je bezdimenzionalnost koja osigurava neovisnost mjere o dimenzijama učenih veličina i mogućnost usporedbe izvedenih algoritama učenja s drugim algoritmima, neovisno o korištenoj sklopovskoj ili programskoj podršci.

4. UNAPRIJEDNA STATIČKA NEURONSKA MREŽA ZA PREDVIĐANJE PONAŠANJA KAOTIČNOG DINAMIČKOG SUSTAVA

Tradicionalne statističke metode obrade signala daju slabe rezultate u predviđanju dinamičkih sustava, a poseban problem su sustavi s nelinearnostima, vremenski promjenjivim parametrima i kaotični sustavi. Kaos ima realnu fizikalnu osnovu, a u izvornom obliku se javlja u kemijskim reaktorima i plazmi, kod turbulencije fluida, lasera itd. Promjenom parametara kaotični sustavi pokazuju širok raspon najrazličitijih nelinearnih ponašanja, te zato predstavljaju zgodan problem za ispitivanje raznih tehnika obrade signala [1].

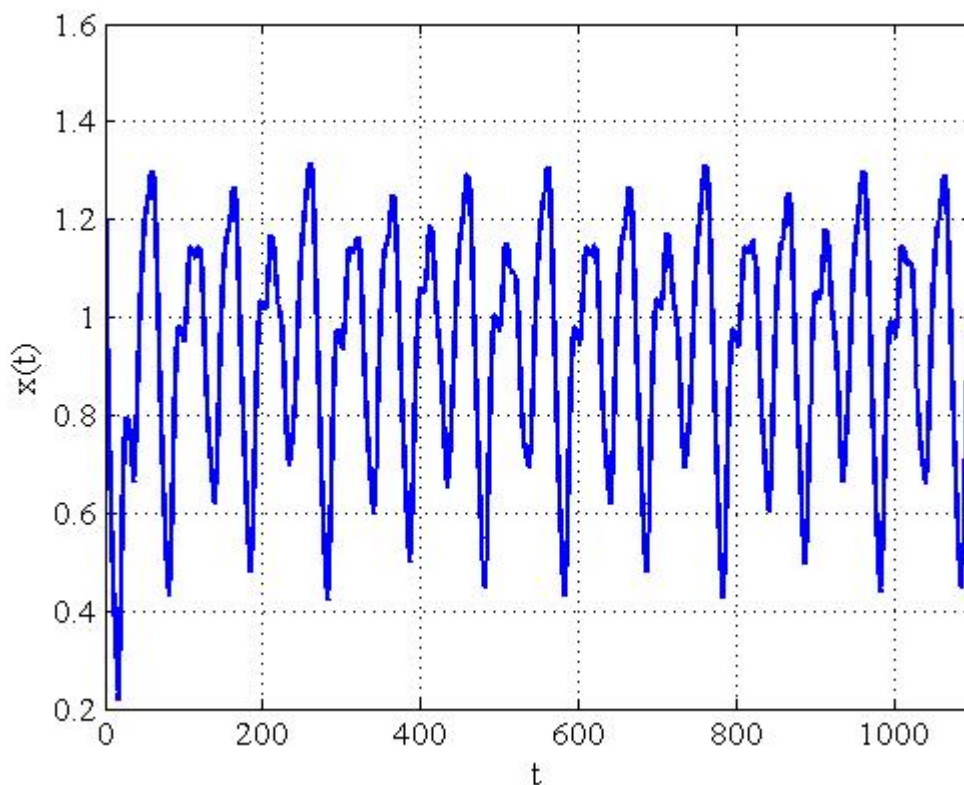
Jedan od tipičnih primjera primjene izvedenih algoritama je obrada signala i predviđanje ponašanja dinamičkih sustava. Izvedeni algoritam za statičku neuronsku mrežu s jednim i s dva sakrivena sloja će se testirati na primjeru predviđanja ponašanja nelinearnog kaotičnog sustava pod nazivom Glass-Mackeyeva jednadžba [8].

4.1. Glass-Mackeyev nelinearni kaotični sustav

Glass-Mackeyev sustav odabran je za testiranje algoritma učenja mreže, zato što je jednostavno definiran, a njegovo ponašanje teško se predviđa. Glass-Mackeyeva jednadžba je nelinearna diferencijalna jednadžba sa kašnjenjem za interval τ , a određena je početnom funkcijom:

$$\dot{x} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \quad (4.1)$$

gdje su a i b parametri sustava, a t je oznaka vremena. Odabrani su parametri $a = 0,2$, $b = 0,1$, te $\tau = 17$, kako bi se dobio stabilan odziv sustava, uz tzv. "fractal" dimenziju $d_A = 2,1$ [1, 7, 8]. Na slici 4.1. prikazan je odziv sustava definiranog izrazom (4.1) za gore odabrane parametre sustava, te uz početni uvjet $x(0) = 1,2$ i periodu testiranja signala $T_0 = 1$ s.



Slika 4.1. Prikaz Glass-Mackeyeve vremenske serije ($\tau = 17$)

4.2. Predviđanje ponašanja kaotičnog sustava

Osnovni zadatak predviđanja ponašanja kaotičnog sustava svodi se na uporabu prošlih vrijednosti vremenskih serija signala $x(n)$ do neke vremenske točke, za predviđanje buduće vrijednosti $x(n+P)$ za P točaka u budućnost [1,7]. Standardna je metoda za ovakav tip predviđanja određivanje funkcije f prema izrazu:

$$x(n+P) = f(x(n), x(n-\Delta), x(n-2\Delta), \dots, x(n-m\Delta)) , \quad (4.2)$$

gdje je P broj točaka za predviđanje u budućnosti, Δ je kašnjenje signala, a m cjelobrojna konstanta. Odabirom $P = \Delta$ moguće je obaviti predviđanje ponašanja sustava P koraka u budućnost. U ovom radu je odabrano $P = 6$. Prema Takens-ovom teoremu [9] raspon vrijednosti za dimenziju d_A iznosi:

$$d_A < m + 1 < 2d_A + 1 \quad (4.3)$$

Za $\tau = 17$ odabrano je $m = 4$. Izraz (4.2) uz odabrane vrijednosti P i m glasi:

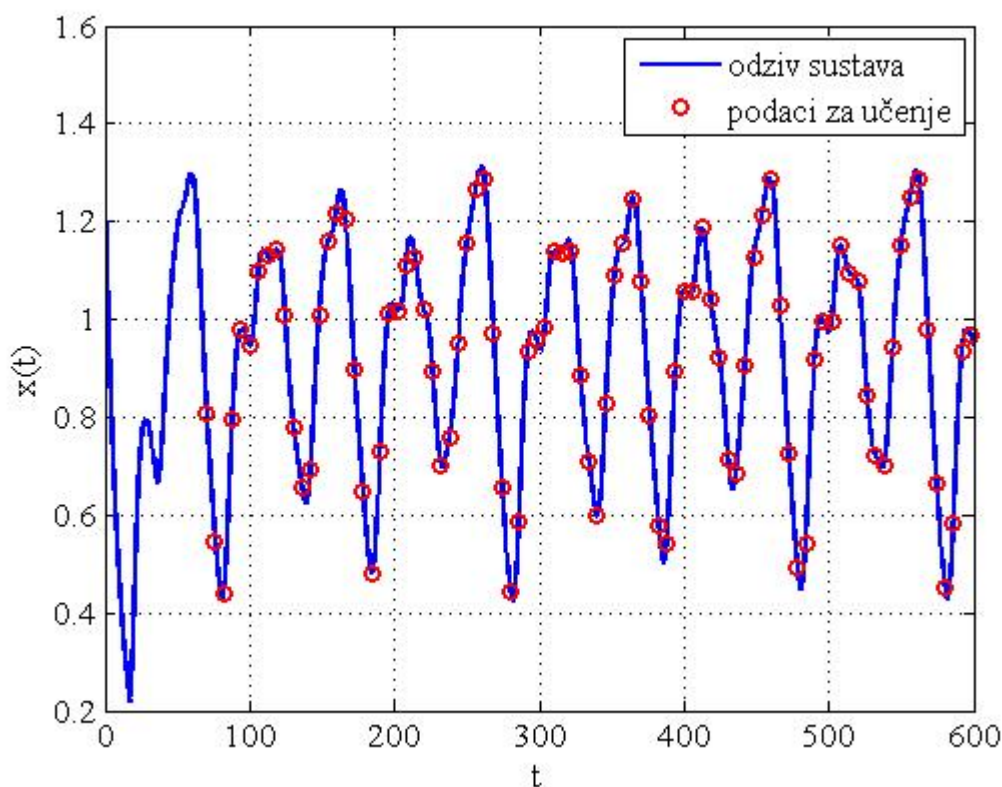
$$x(n+6) = f(x(n), x(n-6), x(n-12), x(n-18), x(n-24)) \quad (4.4)$$

Kako bi se ostvarilo dobro predviđanje šest koraka unaprijed, potrebno je poznavati

formu funkcije f . Pošto je forma te funkcije nepoznata, funkcija f će biti aproksimirana statičkim neuronskim mrežama s jednim i s dva sakrivena sloja, koje će prema izrazu (4.4) imati pet ulaznih (plus BIAS) i jedan izlazni neuron [1]. Zadatak mreža je da na osnovu mjerenih veličina izlaza sustava $x(n)$, $x(n-6)$, $x(n-12)$, $x(n-18)$ i $x(n-24)$ predvide vrijednost izlaza kaotičnog sustava šest koraka unaprijed $x(n+6)$.

4.3. Učenje neuronske mreže

Postupak učenja se provodi na prvih 600 od ukupnog broja snimljenih točaka sustava, s time da je prvih 100 točaka odbačeno zato što se otprilike nakon tog intervala odziv sustava ustabili. Preostalih 500 točaka poslužiti će za testiranje naučene mreže [1]. Na slici 4.2. prikazani su podaci za učenje mreže. Od odziva sustava uzima se svaka šesta vrijednost izlaza sustava x i zajedno s prethodnih četiri diskretizirane vrijednosti dovodi na ulaz neuronske mreže.

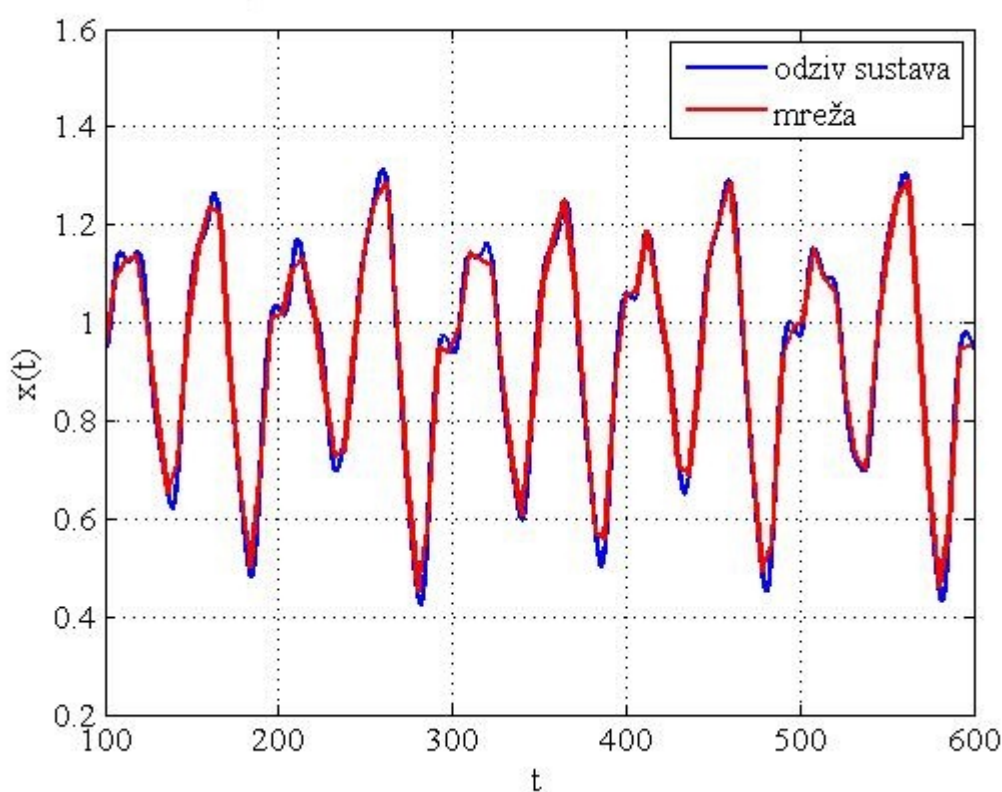


Slika 4.2. Prikaz podataka za učenje mreže

Učenje je provedeno bez momentuma, s momentumom prvog te momentumom drugog reda, te korištenjem sinusne aktivacijske i sigmoidalne aktivacijske funkcije. Prvo će biti prikazano učenje mreže s jednim sakrivenim slojem, a nakon toga s dva sakrivena sloja.

4.3.1. Učenje i testiranje neuronske mreže s jednim sakrivenim slojem

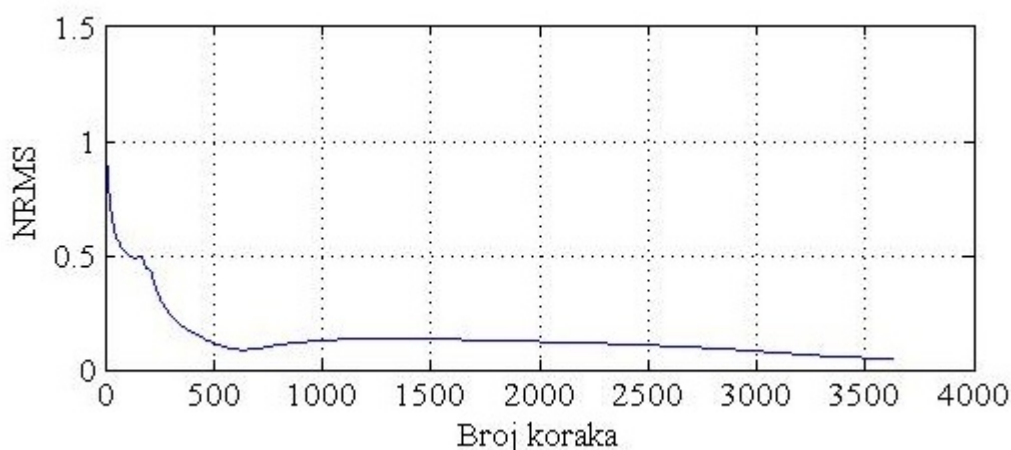
Mreža u sakrivenom sloju ima ukupno 20 neurona. Korišten je koeficijent brzine učenja η iznosa 0,1, momentum prvog reda α iznosa 0,8, sigmoidalna aktivacijska funkcija, te slučajno generirani početni težinski koeficijenti. Učenje se provodilo dok se nije postigao NRMS u iznosu od 0,05 (slika 4.4.).



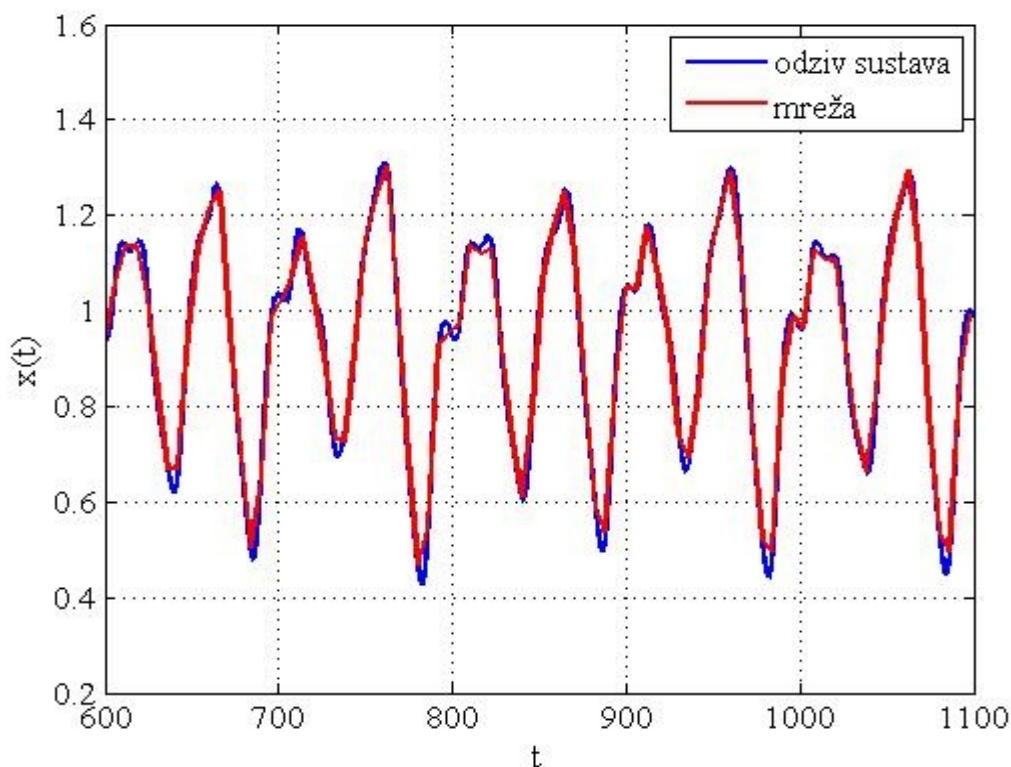
Slika 4.3. Naučena mreža s jednim sakrivenim slojem

Iz slike 4.3. se vidi da je mreža dobro naučila postavljeni zadatak i da na učenim podacima dobro obavlja predviđanje ponašanja nelinearnog kaotičnog sustava za 6 koraka unaprijed, te je ostvaren željeni NRMS (Slika 4.4.). Nakon učenja mreže pokazuje se svojstvo generalizacije naučene mreže, tj. testiranje naučene mreže novim ulaznim

vrijednostima koje mreža nije učila. Test se obavlja na preostalim 500 točaka izlaza sustava (Slika 4.5.). Iz slike 4.5. je vidljivo da naučena mreža dobro generalizira, tj. dobro predviđa ponašanje sustava za njoj nepoznate ulazne vrijednosti (vrijednosti koje nisu obuhvaćene skupom učenja).

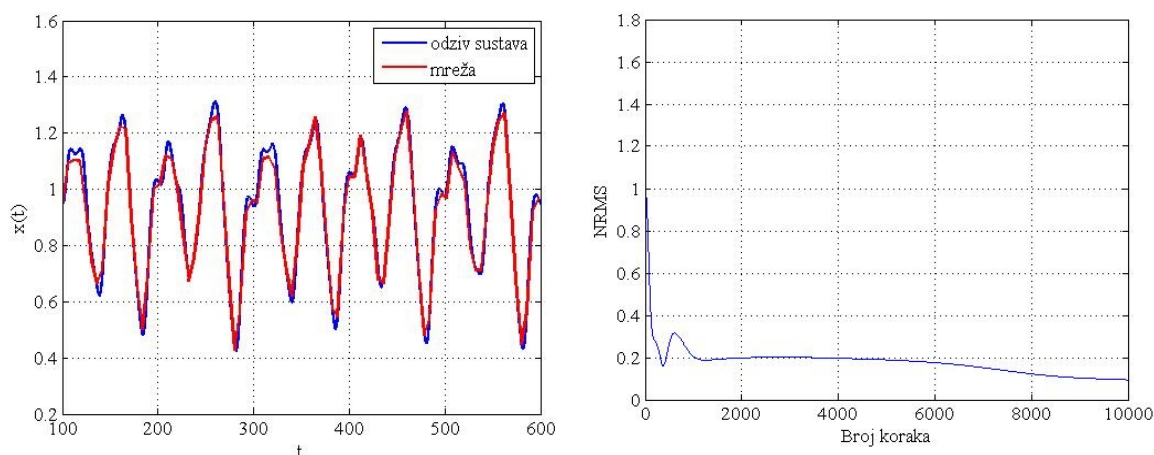


Slika 4.4. Ostvareni NRMS mreže s jednim sakrivenim slojem



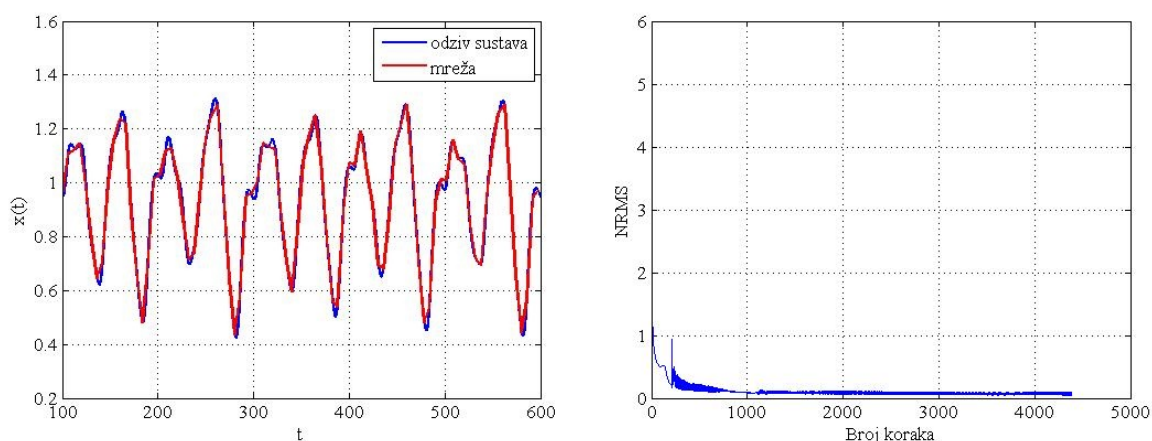
Slika 4.5. Testiranje naučene s jednim sakrivenim slojem mreže na 500 novih i mreži u učenju nepoznatih točaka izlaza sustava

Slijedi prikaz učenja mreže s različitim kombinacijama aktivacijskih funkcija i momentuma. U svim kombinacijama su korišteni jednaki, unaprijed definirani početni težinski koeficijenti kako bi se pokazao utjecaj aktivacijskih funkcija i momentuma na učenje mreže. Koeficijent brzine učenja, željeni NRMS i momentum (kada se koristi) su ostali isti.



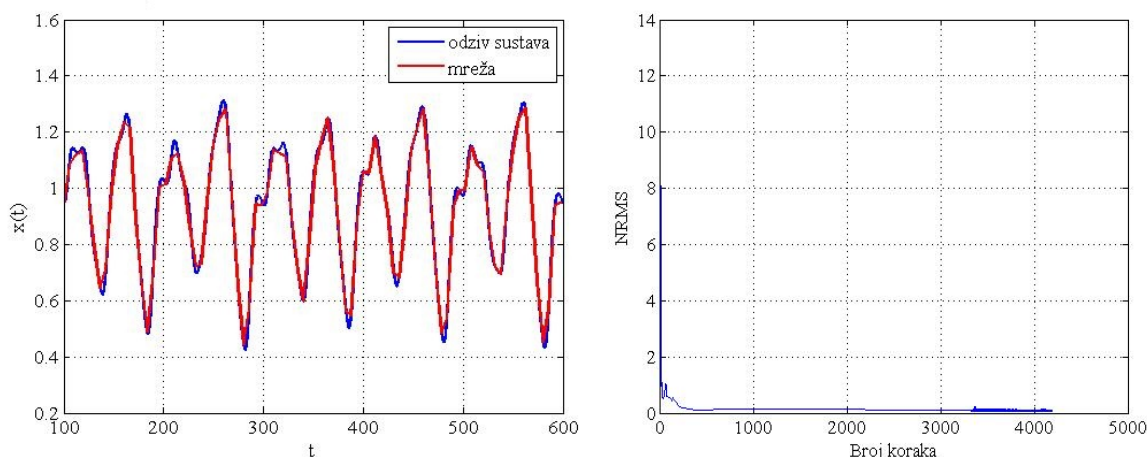
Slika 4.6. Učenje mreže s jednim sakrivenim slojem uz zadanu sigmoidalnu aktivacijsku funkciju bez momentuma

Mreža uz zadanu sigmoidalnom aktivacijskom funkcijom bez momentuma je naučila zadani problem, ali nije ostvaren željeni NRMS (Slika 4.6.). Iz tog razloga se ugrađuje momentum prvog reda kako bi se ubrzao proces učenja. Momentum ubrzava učenje mreže, te se nakon 4389 koraka učenja ostvaruje željeni NRMS (Slika 4.7.).



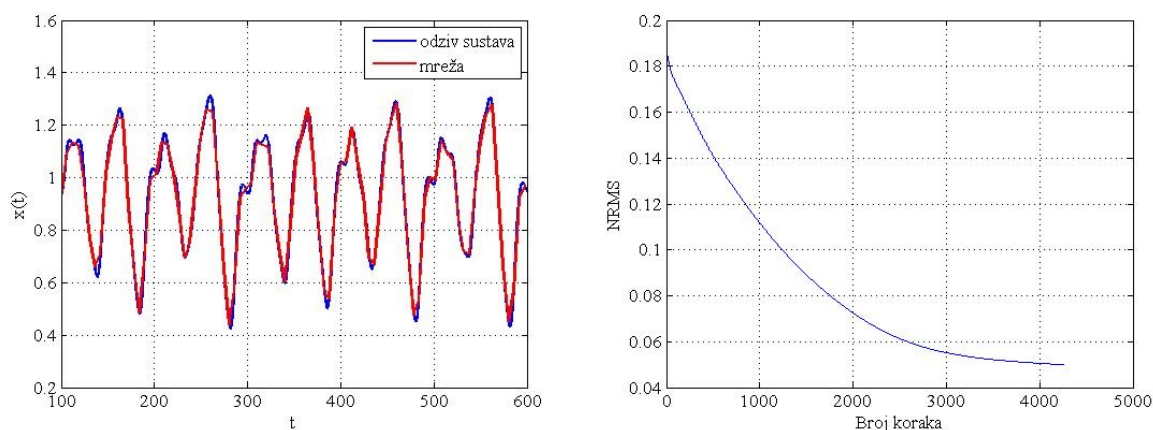
Slika 4.7. Učenje mreže s jednim sakrivenim slojem uz zadanu sigmoidalnu aktivacijsku funkciju i momentum prvog reda

Momentum drugog reda iznosa $\beta = 0,067$, izračunat prema izrazu (2.18), dodatno ubrzava učenje mreže, te se željeni NRMS ostvaruje u 4193 koraku učenja (Slika 4.8.).



Slika 4.8. Učenje mreže s jednim sakrivenim slojem uz zadanu sigmoidalnu aktivacijsku funkciju i momentum drugog reda

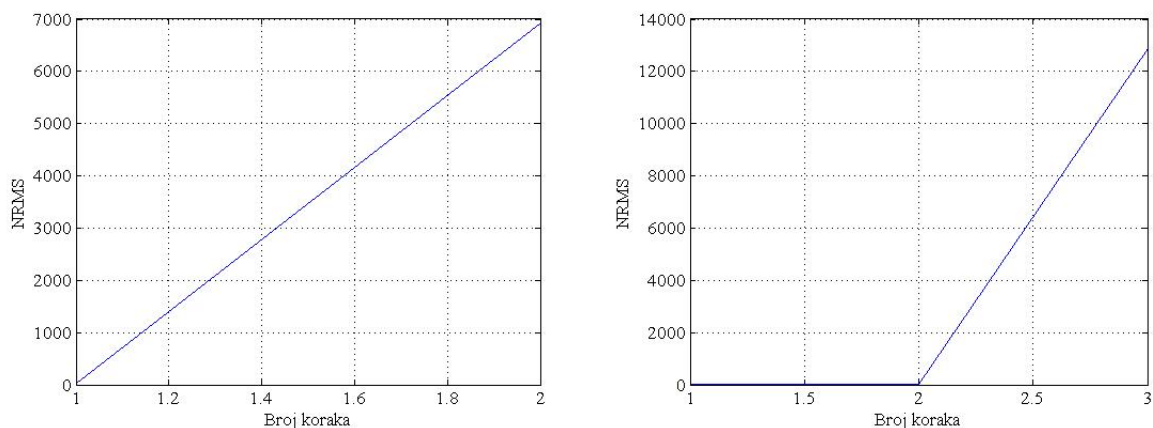
Na slici 4.9. prikazano je učenje mreže uz zadanu sinusnu aktivacijsku funkciju bez momentuma. Mreža je dobro naučila zadani zadatak, te je željeni NRMS ostvaren u 4253 koraku učenja, za razliku od mreže s sigmoidalnom aktivacijskom funkcijom kada nije ostvaren željeni NRMS ni nakon 10 000 koraka učenja.



Slika 4.9. Učenje mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju bez momentuma

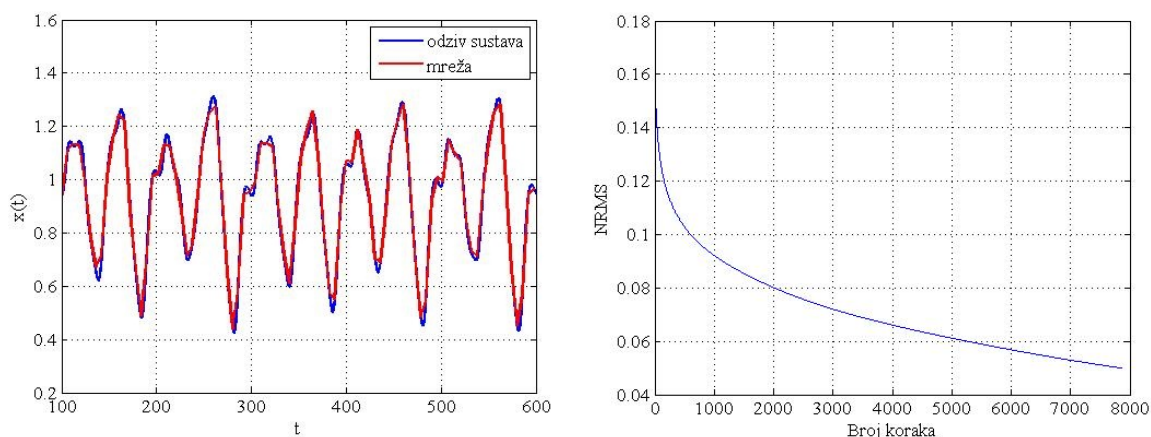
Ugradnjom momentuma prvog i drugog reda u mrežu s sinusnom aktivacijskom funkcijom nije se postiglo brže učenje mreže kao kod mreže s sigmoidalnom aktivacijskom funkcijom, nego željeni NRMS, zbog velike brzine učenja, počinje linearno stalno rasti, te

mreža ne uči. Grafovi ovisnosti NRMS-a o broju koraka učenja mreže s momentumom prvog i drugog reda prikazani su na slici 4.10.

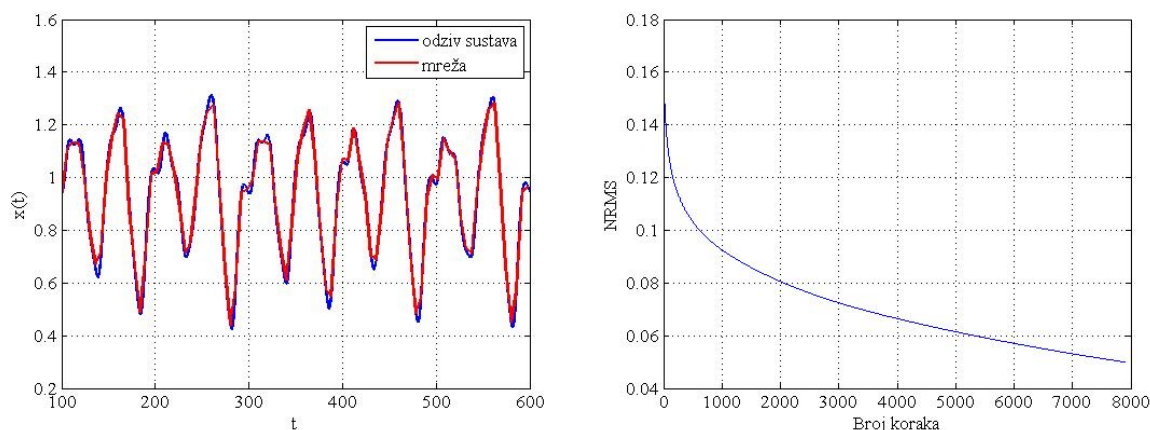


Slika 4.10. NRMS mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju i momentum prvog (lijevo) i drugog reda (desno)

Kako bi se to izbjeglo koeficijent brzine učenja η se smanjuje s 0,1, za jedan red veličine, na 0,01, a svi drugi parametri ostaju identični. Mreža sada za obje kombinacije uči, a željeni NRMS se ostvaruje nakon 7856, odnosno 7905 koraka učenja (Slike 4.11. i 4.12.). Prema dobivenim podacima momentum drugog reda ne ubrzava učenje mreže, nego ga čak malo i usporava. Zbog smanjene brzine učenja mreža puno sporije uči nego bez momentuma uz $\eta = 0,1$, ali se zato ostvaruje konvergencija pri korištenju momentuma.



Slika 4.11. Učenje mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju i momentum prvog reda uz koeficijent brzine učenja $\eta = 0,01$



Slika 4.12. Učenje mreže s jednim sakrivenim slojem uz zadanu sinusnu aktivacijsku funkciju i momentum drugog reda uz koeficijent brzine učenja $\eta = 0,01$

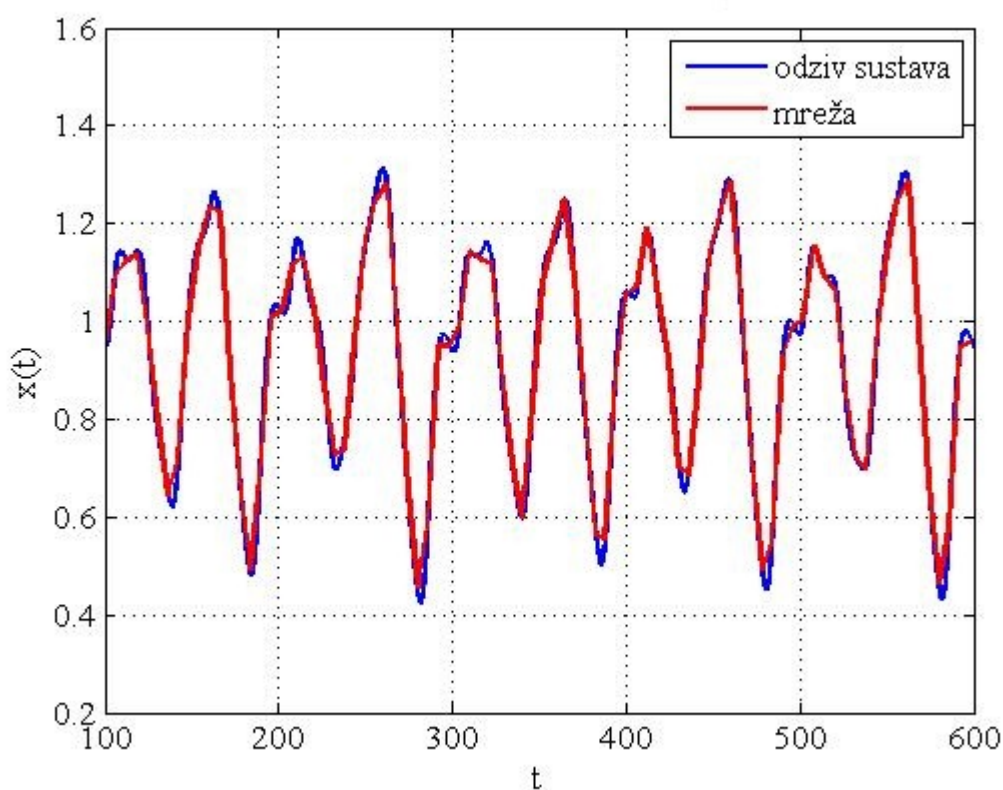
Tablica 4.1. Usporedba učenja mreža s jednim sakrivenim slojem s različitim kombinacijama aktivacijskih funkcija i momentuma

Aktivacijska funkcija	Momentum	Broj koraka učenja za zadani NRMS	Vrijeme učenja [s]
Sigmoidalna	bez	10 000 (NRMS=0.0944)	47.62
	prvog reda	4389	20.56
	drugog reda	4193	19.66
Sinusna	bez	4253	20.30
	prvog reda	7856 ($\eta = 0.01$)	37.63
	drugog reda	7905 ($\eta = 0.01$)	38.18

U tablici 4.1 prikazana je usporedba svih kombinacija aktivacijskih funkcija i momentuma mreže s jednim sakrivenim slojem za zadane parametre učenja i iste početne težinske koeficijente. Prema podacima mreža s sigmoidalnom aktivacijskom funkcijom i momentumom drugog reda najbrže je naučila zadani problem, a ista mreža bez momentuma najsporije, dok je mreža s sinusnom aktivacijskom funkcijom i bez momentuma malo sporije naučila u odnosu na mrežu s sigmoidalnom funkcijom i momentumom drugog reda, ali zato dodavanjem momentuma ista mreža uopće nije učila.

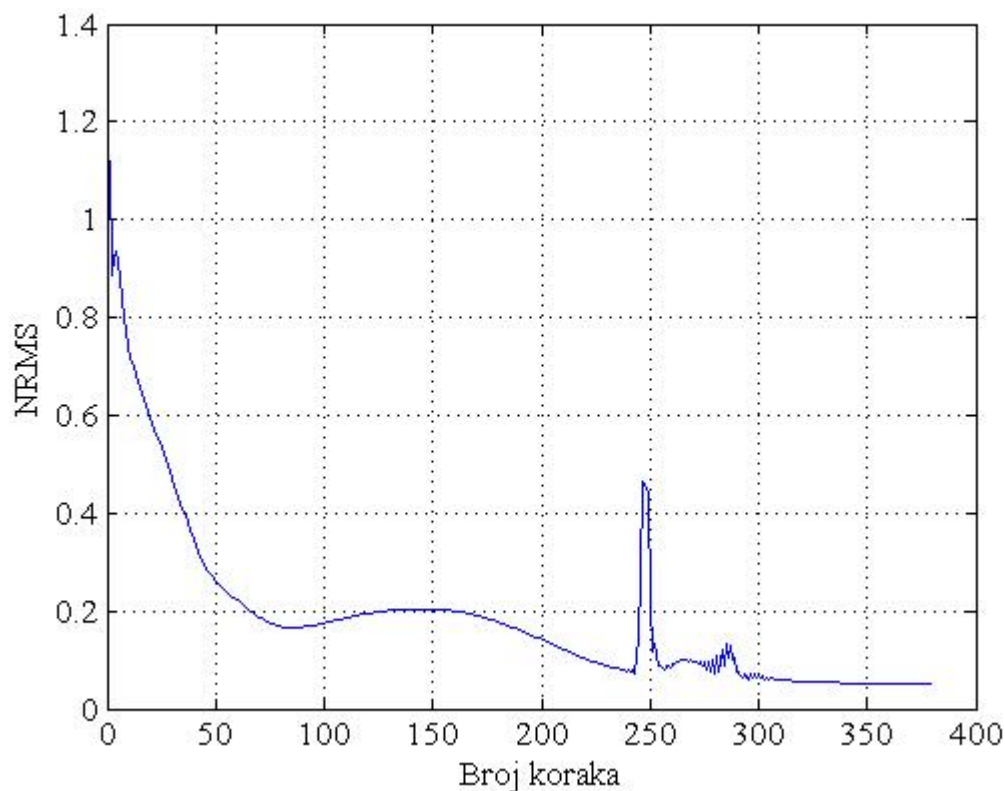
4.3.2. Učenje i testiranje neuronske mreže s dva sakrivena sloja

Mreža u svakom sloju ima po 10 neurona. Korišten su isti parametri učenja kao i kod mreže s jednim sakrivenim slojem. Koeficijent brzine učenja η iznosa 0,1, momentum prvog reda α iznosa 0,8, sigmoidalna aktivacijska funkcija, te slučajno generirani početni težinski koeficijenti. Učenje se provodilo dok se nije postigao zadani NRMS u iznosu od 0,05.

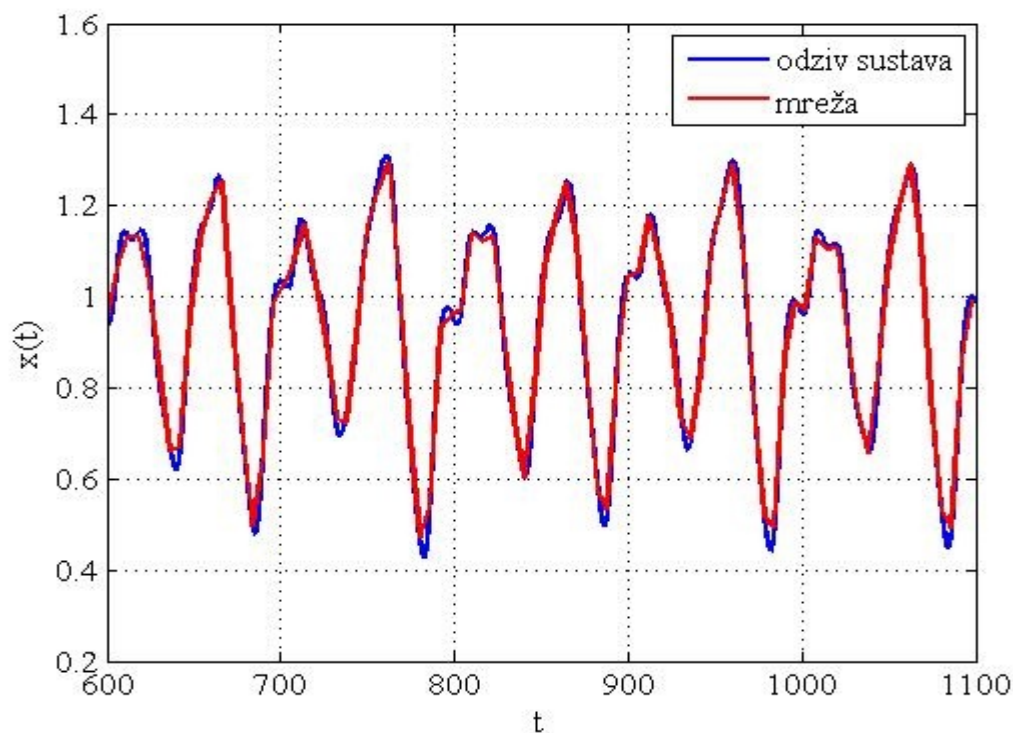


Slika 4.13. Naučena mreža s dva sakrivena sloja

Iz slike 4.13. je vidljivo da je mreža s dva sakrivena sloja također naučila zadani problem te da dobro obavlja predviđanje ponašanja nelinearnog kaotičnog sustava 6 koraka unaprijed, ali za razliku od mreže s jednim sakrivenim slojem, željeni NRMS je ostvaren otprilike deset puta brže (Slika 4.14.). Nakon učenja mreže provodi se test na 500 novih i mreži nepoznatih točaka izlaza sustava (Slika 4.15.), te je vidljivo da naučena mreža dobro generalizira, tj. dobro predviđa ponašanje sustava za njoj nepoznate ulazne.



Slika 4.14. Ostvareni NRMS mreže s dva sakrivena sloja



Slika 4.15. Testiranje naučene mreže s dva sakrivena sloja na 500 novih i mreži u učenju nepoznatih točaka izlaza sustava

Slijedi tablična usporedba učenja neuronske mreže s dva sakrivena sloja za različite kombinacije aktivacijskih funkcija i momentuma. Za sve kombinacije su korišteni jednaki, unaprijed definirani početni težinski koeficijenti. Koeficijent brzine učenja, momentum (kada se koristi), te željeni NRMS ostaju isti.

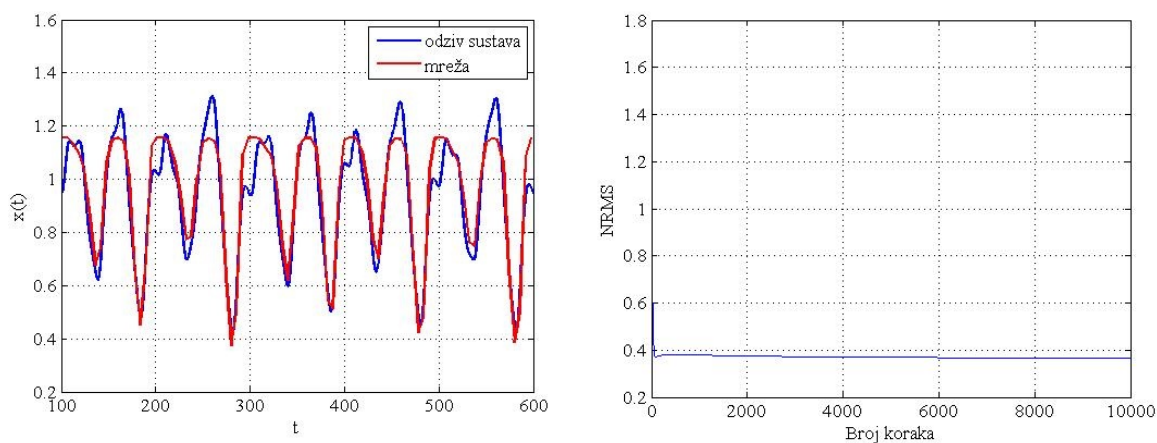
Tablica 4.2. Usporedba učenja mreža s dva sakrivena sloja za različite kombinacije aktivacijskih funkcija i momentuma

Aktivacijska funkcija prvog sakrivenog sloja	Aktivacijska funkcija drugog sakrivenog sloja	Momentum	Broj koraka učenja za zadani NRMS	Vrijeme učenja [s]
sigmoidalna	sigmoidalna	bez	2208	17.11
		prvog reda	581	4.56
		drugog reda	864	6.78
sigmoidalna	sinusna	bez	5020	40.56
		prvog reda	597	4.83
		drugog reda	764	6.16
sinusna	sigmoidalna	bez	1712	13.38
		prvog reda	419	3.31
		drugog reda	256	2.07
sinusna	sinusna	bez	1153	9.41
		prvog reda	199 ($\eta = 0.05$)	1.69
		drugog reda	744 ($\eta = 0.05$)	6.17

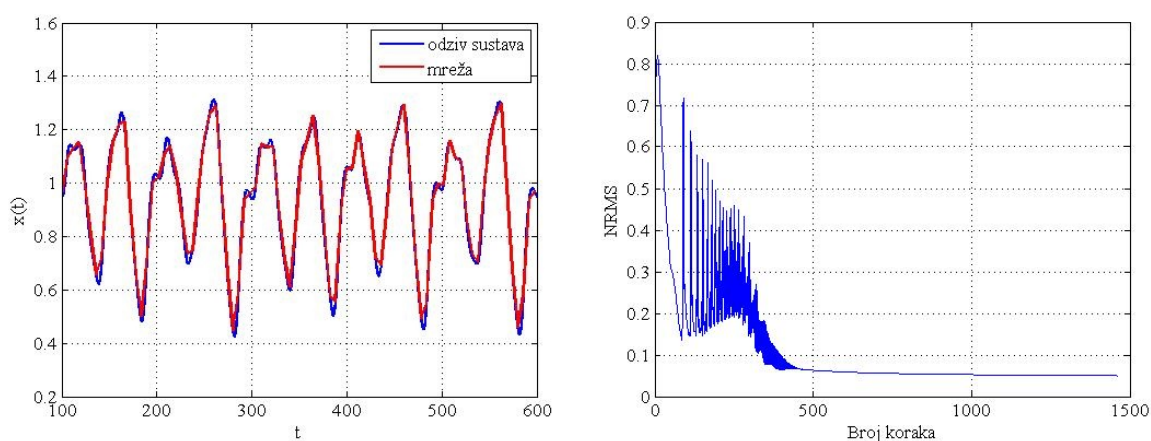
Prema podacima iz tablice 4.2. mreža sa sinusnom aktivacijskom funkcijom prvog sloja i sigmoidalnom aktivacijskom funkcijom drugog sloja, te momentom drugog reda najbrže je naučila zadani zadatak. Pri tome je mreža sa sigmoidalnom aktivacijskom funkcijom prvog sakrivenog sloja i sinusnom aktivacijskom funkcijom drugog sloja, te bez momentuma najsporije naučila zadani zadatak. Kao i kod mreže s jednim sakrivenim slojem, kod mreže s dva sakrivena sloja i sinusnom aktivacijskom funkcijom u oba sakrivena sloja te momentumom prvog i drugog reda NRMS linearno stalno raste i mreža ne uči. Problem se isto kao i kod mreže s jednim sakrivenim slojem rješava smanjivanjem brzine učenja. Uz

upola manju brzinu učenja ($\eta = 0,05$) mreža s aktivacijskim funkcijama u oba sakrivena sloja i momentumom prvog reda postiže veću brzinu učenja od svih ostalih kombinacija.

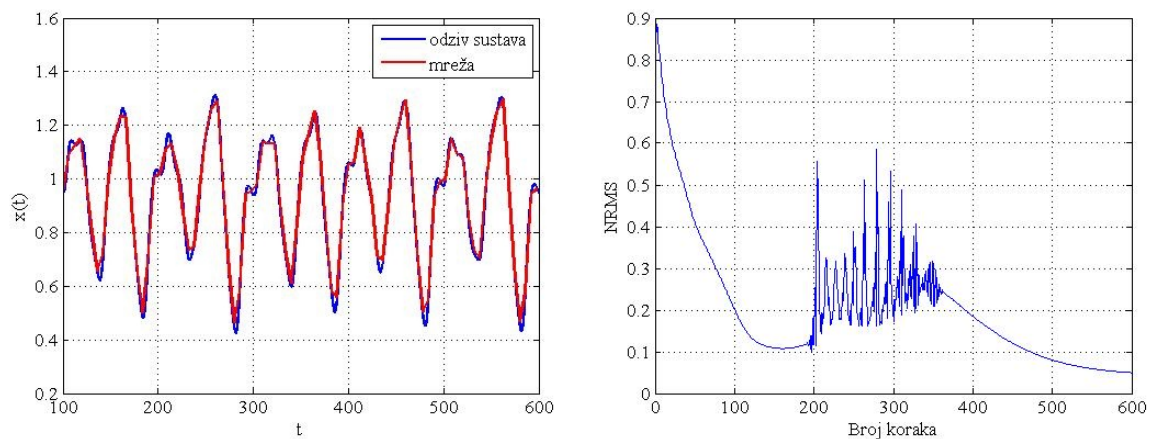
Nakon usporedbe brzina učenja različitih kombinacija aktivacijskih funkcija i momentuma slijedi prikaz učenja mreža s različitim brojem neurona u pojedinom sakrivenom sloju (s ukupno 20 neurona u oba sloja). Radi usporedbe u svim kombinacijama biti će korišteni isti parametri učenja te jednaki početni težinski koeficijenti. Koeficijent brzine učenja iznosi $\eta = 0,1$, momentum prvog reda iznosi $\alpha = 0,8$, željeni NRMS iznosi 0,05, te se koristi sigmoidalna aktivacijska funkcija. Rezultati učenja će biti prikazani i u tablici.



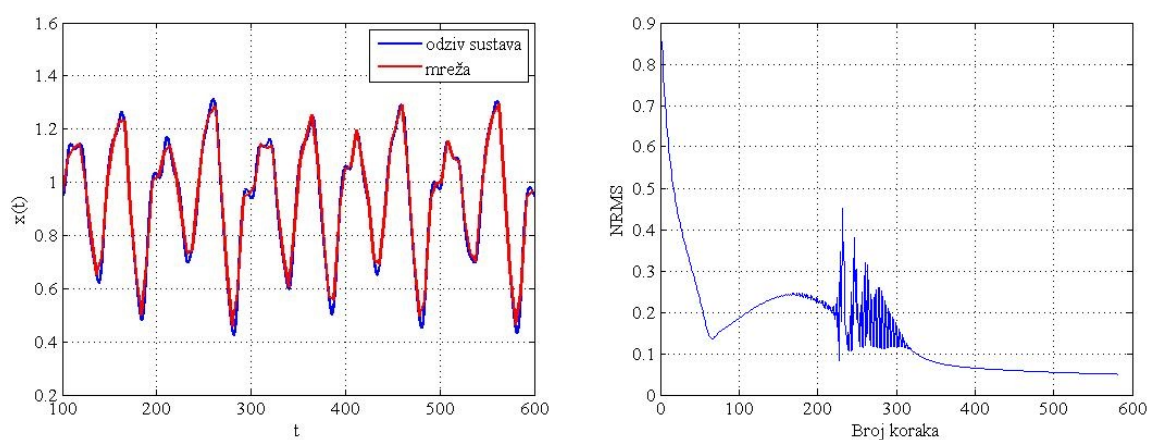
Slika 4.16. Učenje mreže s 2 neurona prvog sloja i 18 neurona drugog sloja



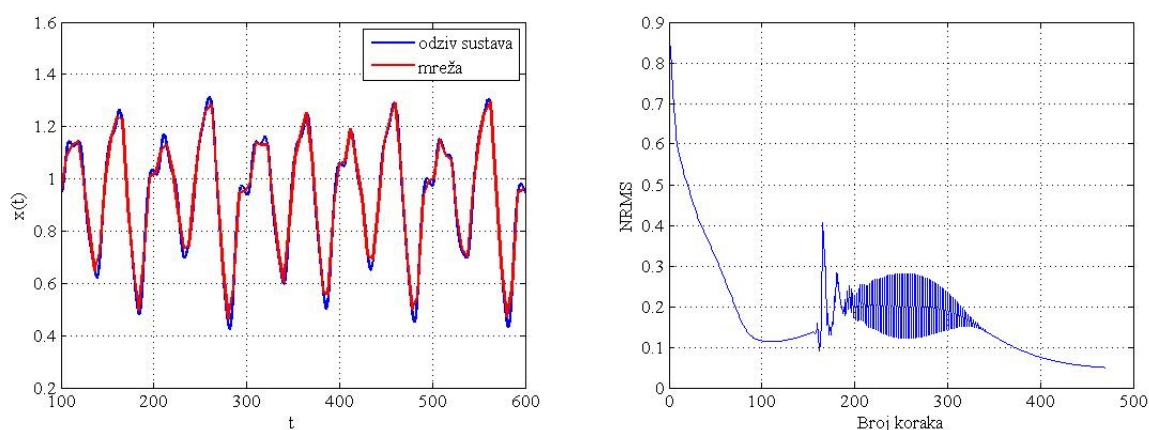
Slika 4.17. Učenje mreže s 5 neurona prvog sloja i 15 neurona drugog sloja



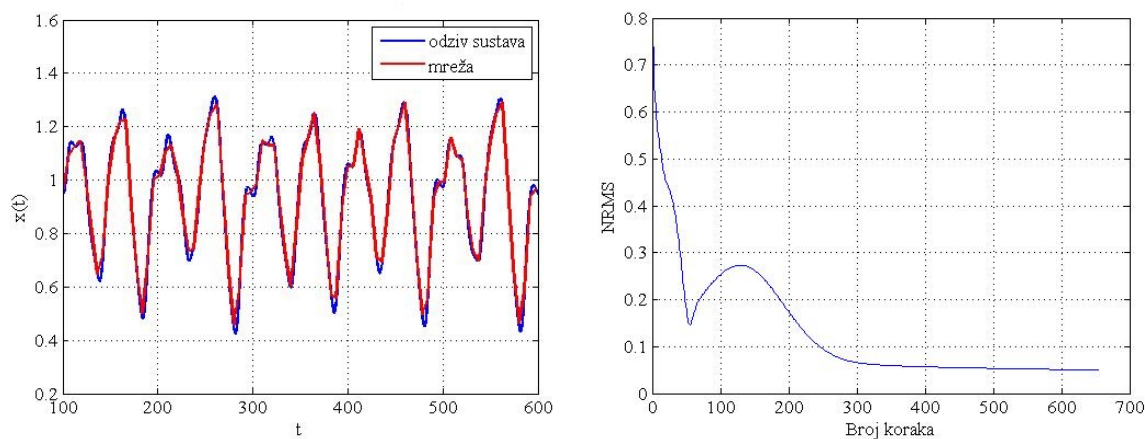
Slika 4.18. Učenje mreže s 8 neurona prvog sloja i 12 neurona drugog sloja



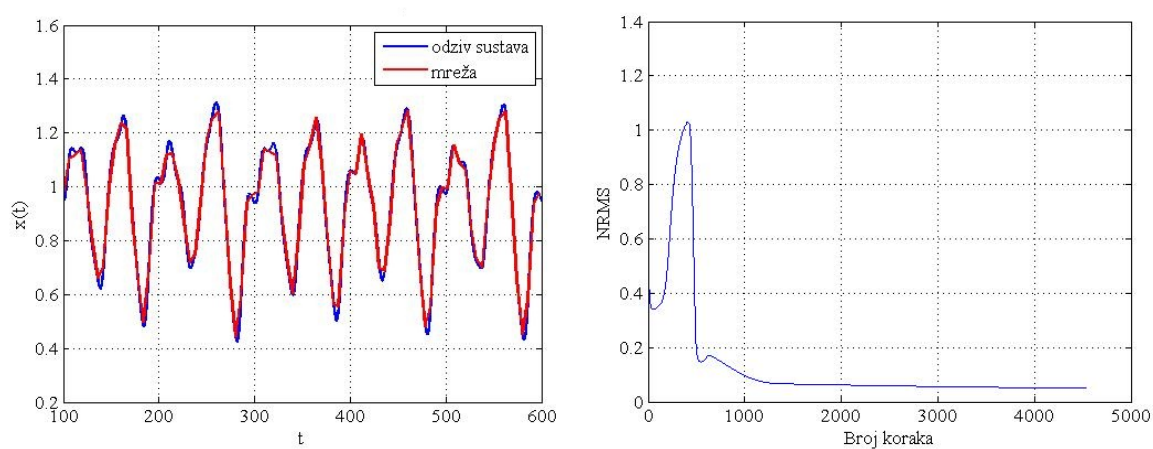
Slika 4.19. Učenje mreže s jednakim brojem neurona sakrivenih slojeva



Slika 4.20. Učenje mreže s 12 neurona prvog sloja i 8 neurona drugog sloja



Slika 4.21. Učenje mreže s 15 neurona prvog sloja i 5 neurona drugog sloja



Slika 4.22. Učenje mreže s 18 neurona prvog sloja i 2 neurona drugog sloja

Prema rezultatima učenja mreže sa svim kombinacijama neurona sakrivenih slojeva, osim s 2 neurona prvog sakrivenog i 18 neurona drugog sakrivenog sloja, su naučile zadani zadatak. Isto tako, kada su neuroni sakrivenih slojeva bili podjednako raspoređeni po oba sloja, mreža je najbrže naučila zadani zadatak.

Tablica 4.3. Usporedba učenja mreža s dva sakrivena sloja za različite kombinacije broja neurona sakrivenih slojeva

Broj neurona prvog sakrivenog sloja	Broj neurona drugog sakrivenog sloja	Broj koraka učenja za zadani NRMS	Vrijeme učenja [s]
2	18	10 000 (NRMS=0.3664)	43.45
5	15	1460	9.26
8	12	599	4.51
10	10	581	4.55
12	8	470	3.66
15	5	654	4.53
18	2	4538	23.51

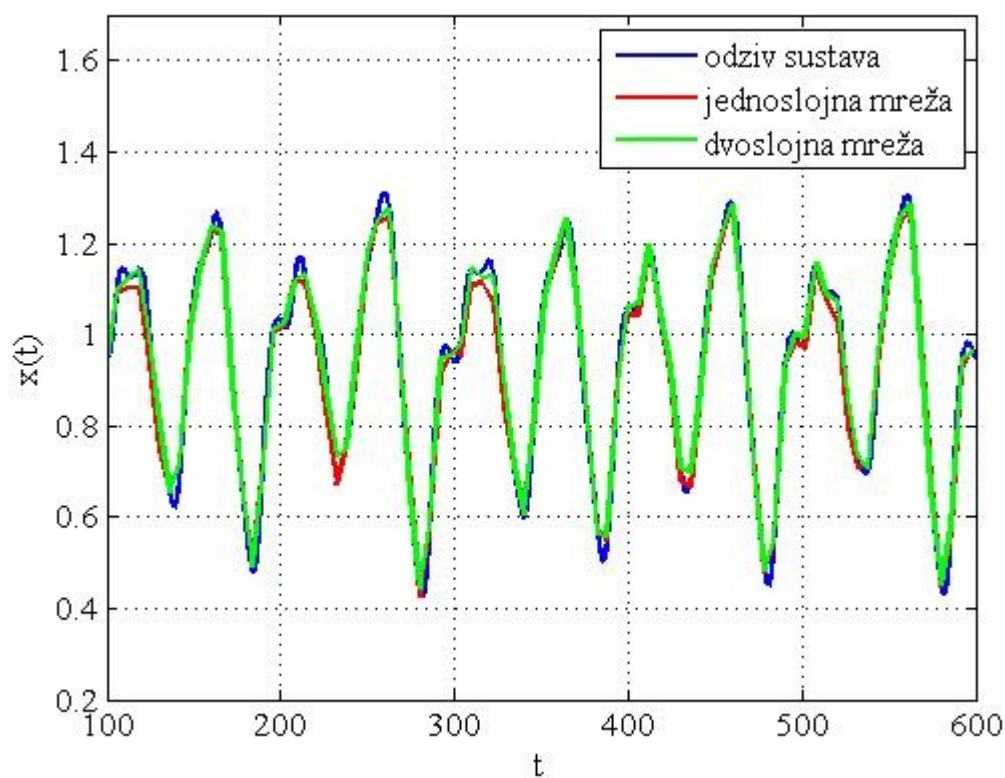
4.3.3. Usporeba učenja i testiranja neuronskih mreža s jednim i s dva sakrivena sloja

Nakon što su uspoređene različite kombinacije aktivacijskih funkcija i momentuma te različite strukture mreža, slijedi usporedba učenja i testiranja mreža s jednim i s dva sakrivena sloja. Obje mreže će imati ukupan broj neurona u sakrivenim slojevima te će biti korišteni isti početni težinski koeficijenti u obje mreže. Koeficijent brzine učenja iznosi $\eta = 0,1$, ne koristi se momentum, željeni NRMS iznosi 0,05 i koristi se sigmoidalna aktivacijska funkcija u obje mreže.

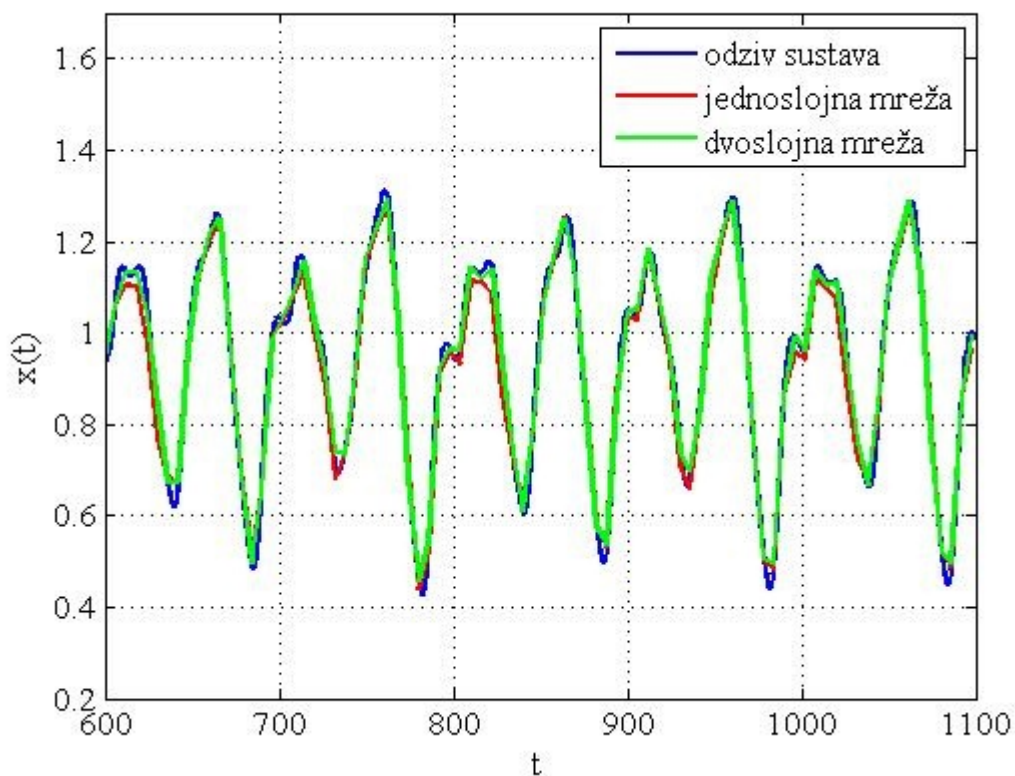
Prema slici 4.23. i prema podacima iz tablice 4.4. vidljivo je da je mreža s dva sakrivena sloja bolje naučila zadani zadatak nego mreža s jednim sakrivenim slojem. Nakon učenja obje mreže su testirane na preostalih 500 točaka izlaza sustava, te prema slici 4.24. mreža s dva sakrivena sloja bolje predviđa ponašanje nelinearnog kaotičnog sustava, čak i za nepoznate ulazne podatke.

Tablica 4.4. Usporedba učenja mreža s jednim i s dva sakrivena sloja

Mreža s:	Broj koraka učenja za zadani NRMS	Vrijeme učenja [s]
jednim sakrivenim slojem	10 000 (NRMS=0.0944)	47.86
dva sakrivena sloja	2208	17.11



Slika 4.23. Usporedba učenja mreža s jednim i s dva sakrivena sloja



Slika 4.24. Usporedba testiranja mreža s jednim i s dva sakrivena sloja

5. ZAKLJUČAK

Cilj ovoga rada je pokazati kvalitetu učenja i generalizacije statičkih neuronskih mreža s povratnim prostiranjem pogreške s jednim i s dva sakrivena sloja na primjeru učenja kaotičnog dinamičkog sustava. Obje mreže, radi usporedbe, imaju isti ukupan broj neurona sakrivenih slojeva te su korištene dvije aktivacijske funkcije neurona sakrivenih slojeva: nelinearna bipolarna sigmoidalna funkcija i sinusna funkcija. U izlaznom sloju neuronske mreže je korištena linearna aktivacijska funkcija kako bi se omogućilo dobivanje rezultata manjih od -1 i većih od 1 na izlazu mreže. Učenje neuronske mreže provodi se s učiteljem. Također, učenje je iterativno te se provodi po uzorku. Zbog velikog broja iteracija prilikom učenja koristi se zamah prvog i drugog reda kako bi se ubrzalo učenje mreže.

Za primjer učenja predikcije ponašanja kaotičnog sustava uzet je Glass-Mackeyev nelinearni kaotični sustav. Nakon snimanja i diskretizacije odziva sustava, definirana je datoteka s podacima za učenje i datoteka s podacima za testiranje naučene mreže. Učenje je prvo provedeno na mreži s jednim sakrivenim slojem, pa zatim na mreži s dva sakrivena sloja. Promatran je utjecaj pojedine aktivacijske funkcije te momentuma na učenje mreže. Za mrežu s dva sakrivena sloja također je promatran utjecaj broja neurona u pojedinom sakrivenom sloju. Pri učenju i testiranju obje mreže korišteni su identični početni težinski koeficijenti sakrivenih i izlaznih slojeva neuronske mreže.

Neuronska mreža s jednim sakrivenim slojem je najbrže naučila zadani zadatak kada je korištena sigmoidalna aktivacijska funkcija neurona sakrivenog sloja i momentum drugog reda. Bez korištenja momentuma mreža je uz obje zadane aktivacijske funkcije naučila zadani zadatak, ali mreža s sigmoidalnom aktivacijskom funkcijom nije postignula željeni NRMS. Korištenjem momentuma uz zadanu sinusnu aktivacijsku funkciju, zbog velike brzine učenja, mreža nije učila. Smanjivanjem koeficijenta brzine učenja mreža je naučila zadani zadatak ali puno sporije nego uz veću brzinu učenja i bez korištenja momentuma, a momentum drugog reda umjesto da ubrza je usporio učenje mreže. Naučena mreža je testirana na novim, njoj nepoznatim ulaznim podacima te je dobro generalizirala i uspješno predviđala ponašanje sustava šest koraka unaprijed.

Kod učenja mreže s dva sakrivena sloja javio se isti problem kao i kod mreže s jednim sakrivenim slojem. Uz korištenu sinusnu aktivacijsku funkciju neurona oba sakrivena sloja s

momentumom mreža nije učila. Smanjenjem brzine učenja taj problem je riješen, ali za razliku od mreže s jednim sakrivenim slojem gdje je učenje bilo puno sporije, ovdje je broj potrebnih iteracija bio puno manji nego za bilo koju drugu kombinaciju aktivacijskih funkcija i momentuma, te je mreža najbrže naučila zadani zadatak. Isto kao i u prethodnom slučaju momentum drugog reda je usporio učenje mreže uz korištenje sinusne aktivacijske funkcije oba sakrivena sloja. Što se tiče broja neurona u pojedinom sakrivenom sloju mreže, mreža je najbrže naučila zadani zadatak kada je broj neurona prvog sakrivenog sloja bio malo veći od broja neurona drugog sakrivenog sloja. Općenito, kada je broj neurona bio podjednak u oba sakrivena sloja, mreža je vrlo brzo postigla željenu točnost. Naučena mreža s dva sakrivena sloja je također testirana na novim, njoj nepoznatim ulaznim podacima te je dobro generalizirala i uspješno predviđala ponašanje sustava šest koraka unaprijed.

Može se zaključiti da je za rješavanje ovakvog tipa zadatka neuronska mreža s dva sakrivena sloja puno bolji izbor od mreže s jednim sakrivenim slojem. Sinusna aktivacijska funkcija je općenito pokazala bolje rezultate pri učenju mreže. Momentum prvog i drugog reda mogu ubrzati proces učenja, ali zato ne garantiraju konvergenciju. Zbog velike brzine učenja korištenjem momentuma dolazi do oscilacija pa mreža ne uči. Podešavanjem koeficijenta brzine učenja taj problem je moguće izbjeći i postići brže učenje neuronske mreže.

LITERATURA

- [1] B.Novaković, D.Majetić, M.Široki: Umjetne neuronske mreže, Zagreb, 1998.
- [2] W.James: Psychology (Briefer course), dostupno na <https://archive.org/details/psychologybriefe00willuoft>
- [3] W.McCulloch, W.Pitts: A logical calculus of the ideas immanent in nervous activity, dostupno na: http://www.aemea.org/math/McCulloch_Pitts_1943.pdf
- [4] <http://en.wikipedia.org/wiki/Neuron>
- [5] http://en.wikipedia.org/wiki/Artificial_neuron
- [6] http://en.wikipedia.org/wiki/Artificial_neural_network
- [7] A.Lapedes, R.Farber: How neural nets work, dostupno na: <http://papers.nips.cc/paper/59-how-neural-nets-work.pdf>
- [8] http://www.scholarpedia.org/article/Mackey-Glass_equation
- [9] http://en.wikipedia.org/wiki/Takens%27_theorem